Correction du devoir de probabilités

Problème :

Une urne contient $2n$ jetons numérotés de 1 à n : chaque jeton apparaît deux fois dans l'urne sous deux couleurs distinctes (bleue et rouge). On effectue des tirages simultanés de deux jetons. Si les jetons tirés sont identiques (en valeur) alors on les enlève de l'urne. Sinon, on les remet tous les deux dans l'urne.
Soit T_n la variable aléatoire qui compte le nombre de tirages nécessaires pour vider l'urne de son contenu.

① Cas où l'urne ne contient qu'une paire ($n = 1$) : On demande quelle loi suit la variable T_1.
Il suffit de dire qu'au premier tirage, on sortira nécessairement l'unique paire présente et donc T_1 est une variable aléatoire certaine égale à 1.
Conclusion : $\boxed{T_1(\Omega) = \{1\} \text{ et } \mathbb{P}(T_1 = 1) = 1}$

② On suppose $n = 2$: (2 paires). Soit P_i l'événement : « on tire une paire au tirage n° i ».

a) *Déterminons $T_2(\Omega)$:* On ne peut pas extraire les deux paires de l'urne avant le deuxième tirage. A l'inverse, il est possible de tirer systématiquement deux jetons non appariés avant d'obtenir y-compris la première paire...
Conclusion : $\boxed{T_2(\Omega) = [\![2, +\infty[\![}$

b) *Combien y a-t-il de tirages simultanés possibles de 2 jetons ?* Il s'agit d'une 2-combinaison d'un ensemble à 4 éléments (1 et 2 rouges, 1 et 2 bleus). D'où, $\text{Card}(\Omega) = \dbinom{4}{2} = 6$.

Dès lors, les tirages étant équiprobables, on a : $\mathbb{P}(P_1) = \dfrac{2}{6}$ car il y a deux paires qui constituent un cas favorable.

Conclusion : $\boxed{\mathbb{P}(P_1) = \dfrac{1}{3}}$

c) *Exprimons $(T_2 = 2)$ à l'aide de P_1 et P_2 :* Il est immédiat que $(T_2 = 2) = P_1 \cap P_2$
Et plus généralement,

$$\forall k \geq 2,\ (T_2 = k) = \overline{P_1} \cap \overline{P_2} \cdots \cap \overline{P_{k-2}} \cap P_{k-1} \cap P_k$$

car dès qu'un paire a été extraite, il n'en reste plus qu'une dans l'urne et l'extraire devient un événement certain (autrement dit, on a vidé l'urne au k-ième tirage si, et seulement si, la première paire a été extraite au $(k-1)$-ième tirage).

d) *Montrons que $\mathbb{P}(T_2 = k) = \dfrac{1}{2}\left(\dfrac{2}{3}\right)^{k-1}$ pour tout $k \geq 2$:*

Les épreuves ne sont pas indépendantes puisque, dans le cas où une paire est choisie, elle est extraite de l'urne dont la composition est donc modifiée.
Nous sommes donc contraints d'appliquer la formule des probabilités composées, à savoir :

$$\mathbb{P}(T_2 = k) = \mathbb{P}(\overline{P_1}) \cdot \mathbb{P}_{\overline{P_1}}(\overline{P_2}) \cdots \mathbb{P}_{\overline{P_1} \cap \cdots \cap \overline{P_{k-3}}}(\overline{P_{k-2}}) \cdot \mathbb{P}_{\overline{P_1} \cap \cdots \cap \overline{P_{k-2}}}(P_{k-1}) \cdot \mathbb{P}_{\overline{P_1} \cap \cdots \cap \overline{P_{k-2}} \cap P_{k-1}}(P_k)$$

Or

$$\mathbb{P}_{\overline{P_1} \cap \cdots \cap \overline{P_{i-1}}}(\overline{P_i}) = \frac{2}{3} \text{ pour tout } i \geq 2$$

$$\mathbb{P}_{\overline{P_1} \cap \cdots \cap \overline{P_{k-2}}}(P_{k-1}) = \frac{1}{3}$$

et (il ne reste plus qu'une paire !)

$$\mathbb{P}_{\overline{P_1}\cap\cdots\cap\overline{P_{k-2}}\cap P_{k-1}}(P_k) = 1$$

Conclusion : $\boxed{\mathbb{P}(T_2 = k) = \left(\dfrac{2}{3}\right)^{k-2} \cdot \dfrac{1}{3} \cdot 1 = \left(\dfrac{2}{3}\right)^{k-1} \cdot \dfrac{1}{2}}$

e) *Rappel :* Si $\sum k\mathbb{P}(T_2 = k)$ cv, alors $\mathbb{E}(T_2)$ existe et vaut : $\displaystyle\sum_{k\in T_2(\Omega)} k\mathbb{P}(T_2 = k)$.

Que vaut $E(T_2)$? La série $\displaystyle\sum_{k\geq 2} \frac{1}{2}k\left(\frac{2}{3}\right)^{k-1}$ converge car elle est de même nature que la série géométrique dérivée de raison $q = \dfrac{2}{3} \in]-1, 1[$ (la multiplication par $\lambda = \dfrac{1}{2}$ ne change pas la nature de la série...).
Donc $\mathbb{E}(T_2)$ existe et pour la somme on note qu'elle commence à 2 et non pas à 1. Dès lors :

$$\boxed{\mathbb{E}(T_2) = \frac{1}{2}\left(\frac{1}{(1-\frac{2}{3})^2} - 1\right) = \frac{1}{2}(9-1) = 4}$$

③ On revient au cas n quelconque.

a) *Écrivons une fonction Python qui prend en argument un entier n et qui retourne la liste des $2n$ jetons :* Plusieurs choix s'offrent à nous. En voici quelques exemples qui vont conditionner notre façon de répondre à la question suivante !

— **Choix 1 :** Pour distinguer les couleurs, on modélise les jetons bleus par leur numéro de 1 à n et les jetons rouges par les entiers de $n+1$ à $2n$. Avec ce choix, les couples $(1, n+1), \cdots (n, 2n)$ sont des paires.

```python
def creationListe1(n: int) -> List[int]:
    return [k for k in range(1,2*n+1)]
```

Exemple : `L1 = creationListe1(3)` retourne `L1 = [1, 2, 3, 4, 5, 6]`

— **Choix 2 :** On modélise chaque jeton par un couple (j, c) dont la première coordonnée est égale à son numéro et sa seconde coordonnée est égale à sa couleur. Avec ce choix, pour tout $i \in [\![1, n]\!]$, `(i, 'b')` et `(i, 'c')` est une paire

```python
def creationListe2(n: int) -> List[Tuple[int, int]]:
    L = [k for k in range(1, n+1)]
    C = ['r', 'b']
    return [(j, c) for j in L for c in C]
```

Exemple : `L2 = creationListe2(3)` retourne `L1 = [(1, 'r'), (1, 'b'), (2, 'r'), (2, 'b'), (3, 'r'), (3, 'b')]`

— **Choix 3 :** On modélise chaque jeton par son numéro. Avec ce choix, il y aura 2 représentant de chaque numéro.

```python
def creationListe3(n: int) -> List[int]:
    return [k for k in range(1, n+1)]*2
```

ou encore :

```python
def creationListe3b(n: int) -> List[int]:
    return [k for k in range(1, n+1) for i in range(2)]
```

Exemple : `L3 = creationListe3(3)` retourne `L1 = [1, 2, 3, 1, 2, 3]` et `L3 = creationListe3b(3)` retourne `L1 = [1, 1, 2, 2, 3, 3]`

b) *Écrivons une fonction qui simule le tirage de deux jetons, renvoyant* `True` *si une paire est obtenue et* `False sinon` *:* On va écrire trois fonctions possibles, selon le choix de représentation de l'urne retenu à la question précédente.

— **Choix 1 :** On rappelle que `L = [1, 2, ..., n, n+1, ..., 2*n]` : On utilise la fonction `sample(L, 2)` de la bibliothèque `random` qui permet d'extraire 2 valeurs distinctes prises dans `L`. Soit `(j1, j2)` l'issue d'un tel tirage. On rappelle qu'il s'agit d'une paire si `j2 = j1+n` ou bien si `j1 = j2+n`.

```python
def tiragePaire1(L: List[int]) -> bool:
    n = len(L)//2
    j1, j2 = rdm.sample(L, 2)
    return j2-j1 == n or j1-j2 == n
```

— **Choix 2 :** On rappelle que `L = [(1, 'r'), (1, 'b'), ..., (n, 'b')]` : On utilise ici aussi la fonction `sample(L, 2)` de la bibliothèque `random` qui permet d'extraire 2 valeurs distinctes prises dans `L`. Soit `(j1, j2)` l'issue d'un tel tirage. Il s'agit d'une paire si les numéros coïncident, à savoir si `j1[0] == j2[0]`.

```python
def tiragePaire2(L: List[Tuple[int, int]]) -> bool:
    j1, j2 = rdm.sample(L, 2)
    return j1[0] == j2[0]
```

— **Choix 3 :** On rappelle que `L = [1, 2, ..., n, 1, 2, ..., n]` : On utilise la fonction `rdm.choice(L)` qui retourne un élément pris au hasard (selon une loi uniforme) dans la liste `L` puis on supprime cet élément de la liste avant le second tirage grâce à `L.remove(a)`.
✐ *Remarque :* On prendra soin de travailler sur une copie de `L` pour ne pas perdre notre liste initiale car si l'issue du tirage n'est pas une paire, il faudra à nouveau réaliser un tirage dans cette même urne.
On considérera qu'on a une paire si les deux numéros coïncident.

```python
def tiragePaire3(L: List[int]) -> bool:
    L1 = list(L)
    j1 = rdm.choice(L1)
    L1.remove(j1)
    j2 = rdm.choice(L1)
    return j1 == j2
```

✐*Remarque 2 :* On pouvait aussi travailler sur les indices des jetons (de 1 à $2n-1$) mais la rédaction devient plus compliquée. On utilisera dans ce cas `L.pop(i)` qui supprime le terme d'indice `i` dans la liste `L`.

```python
def tiragePaire3bis(L: List[int]) -> bool:
    L1 = list(L)
    i1 = rdm.randint(0, len(L)-1)
    L1.pop(i1)
    i2 = rdm.randint(0, len(L1)-1)
    return L[i1] == L1[i2]
```

c) *Écrivons une fonction Python* `rgPremierePaire(n)` *d'argument n qui retourne le rang du tirage d'une première paire :*
Ici, peu importe le mode de représentation de l'urne puisqu'on va s'appuyer sur la fonction `tiragePaire()` écrite à la question précédente.

```python
def rgPremierePaire(n: int) -> int:
    L = creationListe(n)
    k = 1
    while not tiragePaire(L):
        k += 1
    return k
```

d) *Écrivons une fonction Python* `simulT(n)` *d'argument n qui simule la variable T_n, autrement dit qui retourne le nombre de tirages nécessaires pour vider une urne composée de n paires :*

Sachant que l'urne est composée de n paire, il faudra faire appel n fois à la fonction `rgPremierePaire()` pour vider l'urne. On utilise pour cela une structure répétitive « Pour » qui appellera successivement la fonction précédente avec, au départ n paires, puis $n-1$ paires, $n-2$, etc. jusqu'à ce qu'il ne restes plus qu'une seule paire dans l'urne, auquel cas l'événement « Tirer une paire » est un événement certain.
Il ne restera plus qu'à retourner la somme de tous les tirages nécessaires pour obtenir chacune des paires et pour ça on introduit un compteur nT qui sera initialisé à 0.

```python
def simulT(n: int) -> int:
    nT = 0
    for k in range(n,0,-1):
        nT += rgPremierePaire(k)
    return nT
```

e) Sachant qu'on nous autorise à faire appel à la fonction `np.mean()` de la bibliothèque `numpy`, il est possible d'évaluer $\mathbb{E}(T_2)$ en appelant un grand nombre de fois la fonction `simulT()` (par exemple 1000 fois) pour un nombre de couples initial de jetons égale à $n = 2$ et en calculant la moyenne des valeurs retournées, soit

`np.mean([simulT(2) for i in range(1000)])`

f) En évaluant $\mathbb{E}(T_n)$ pour $1 \leq n \leq 10$ (avec $m = 1000$ répétitions indépendantes) on a obtenu les valeurs suivantes : `[1, 4.038,8.981,16.293,24.913,36.476,49.7646,63.766,80.7876,99.729]`.
Au regard des valeurs obtenues, on peut conjecturer une formule du type $\boxed{\mathbb{E}(T_n) = n^2}$.

④ On considère l'événement C : « on tire deux jetons identiques lors du premier tirage ».

a) *Montrons que* $\mathbb{P}(C) = \dfrac{1}{2n-1}$ *:* On rappelle que $\text{Card}(\Omega) = \dbinom{2n}{2}$ car il s'agit de dénombrer les 2-combinaisons d'un ensemble comprenant $2n$ éléments.

Or il y a n paires possibles. D'où : $\mathbb{P}(C) = \dfrac{n}{\dbinom{2n}{n}} = \dfrac{n}{\frac{2n(2n-1)}{2}}$.

Conclusion : $\boxed{\mathbb{P}(C) = \dfrac{1}{2n-1}}$.

b) *Après avoir préciser avec quel système complet d'événement vous travaillez, montrons grâce à la formule des probabilités totales que :* $\mathbb{P}(T_n = k+1) = \dfrac{1}{2n-1}\mathbb{P}(T_{n-1} = k) + \dfrac{2n-2}{2n-1}\mathbb{P}(T_n = k)$

Si on se fie à la question précédente, on prendra assez naturellement comme système complet d'événement le système : $\{C, \overline{C}\}$.
Dès lors, d'après la formule des probabilités totales :

$$\mathbb{P}(T_n = k+1) = \mathbb{P}(C)\mathbb{P}_C(T_n = k+1) + \mathbb{P}(\overline{C})\mathbb{P}_{\overline{C}}(T_n = k+1)$$

Or :

— $\mathbb{P}_C(T_n = k+1) = \mathbb{P}(T_{n-1} = k)$: En effet, sachant qu'une paire a été extraite dès le premier tirage, il ne reste plus qu'à vider l'urne désormais composée de $n-1$ paire en k tirages.

— $\mathbb{P}_{\overline{C}}(T_n = k+1) = \mathbb{P}(T_n = k)$: En effet, sachant qu'on n'a pas extrait de paire au premier tirage, il reste n paire dans l'urne et k tirages pour la vider.

Conclusion : $\boxed{\mathbb{P}(T_n = k+1) = \frac{1}{2n-1}\mathbb{P}(T_{n-1} = k) + \frac{2n-2}{2n-1}\mathbb{P}(T_n = k)}$

c) *Déduisons-en que si T_n admet une espérance, alors $\mathbb{E}(T_n) = \mathbb{E}(T_{n-1}) + 2n - 1$:*
Comme $T_n(\Omega) = [\![n, +\infty[\![$, nous supposons que $\sum k\mathbb{P}(T_n = k)$ converge et $\mathbb{E}(T_n) = \sum_{k=n}^{\infty} k\mathbb{P}(T_n = k)$.
Dès lors :

$$\begin{aligned}
\mathbb{E}(T_n) &= \frac{1}{2n-1}\sum_{k=n}^{\infty} k\mathbb{P}(T_{n-1} = k-1) + \frac{2n-2}{2n-1}\sum_{k=n}^{\infty} k\mathbb{P}(T_n = k-1) \\
&= \frac{1}{2n-1}\sum_{k'=n-1}^{\infty} (k'+1)\mathbb{P}(T_{n-1} = k') + \frac{2n-2}{2n-1}\sum_{k'=n-1}^{\infty} (k'+1)\mathbb{P}(T_n = k') \\
&= \frac{1}{2n-1}\left(\sum_{k'=n-1}^{\infty} k'\mathbb{P}(T_{n-1} = k') + \sum_{k'=n-1}^{\infty} \mathbb{P}(T_{n-1} = k')\right) + ... \\
&\qquad + \frac{2n-2}{2n-1}\left(\sum_{k'=n-1}^{\infty} k'\mathbb{P}(T_n = k') + \sum_{k'=n-1}^{\infty} \mathbb{P}(T_n = k')\right) \\
&= \frac{1}{2n-1}(\mathbb{E}(T_{n-1}) + 1) + \frac{2n-2}{2n-1}(\mathbb{E}(T_n) + 1) \text{ car on note que } \mathbb{P}(T_n = n-1) = 0
\end{aligned}$$

Il suffit de passer $\mathbb{E}(T_n)$ à gauche de l'égalité pour obtenir :

$$\frac{1}{2n-1}\mathbb{E}(T_n) = \frac{1}{2n-1}\mathbb{E}(T_{n-1}) + 1$$

Conclusion : $\boxed{\mathbb{E}(T_n) = \mathbb{E}(T_{n-1}) + 2n - 1}$

d) *En quoi cela semble-t-il cohérent avec les résultats du 3.d) ?*
On sait que $\mathbb{E}(T_1) = 1$ puisque T_1 est une variable aléatoire certaine égale à 1.
On en déduit que $\mathbb{E}(T_2) = 4$.
On fait l'hypothèse que $\mathbb{E}(T_n) = n^2$ conformément à la conjecture faire à la question 3.d).
Alors, $\mathbb{E}(T_{n+1} = n^2 + 2(n+1) - 1 = (n+1)^2$.
Conclusion : $\boxed{\forall n \geq 1, \, \mathbb{E}(T_n) = n^2}$ - Cohérent avec la conjecture de 3.d.