Protocoles de calcul à base de cartes

On considère des cartes dont la face recto porte un symbole noté $\boxed{\heartsuit}$ (blanc) ou $\boxed{\spadesuit}$ (noir). Une carte face verso affiche le symbole $\boxed{?}$. Une séquence de cartes est un vecteur $(\alpha_0, \dots, \alpha_{m-1})$ où α_i est formellement une paire indiquant le symbole de la carte (blanc ou noir) et son état (recto ou verso). Dans la suite de l'énoncé, on écrira simplement des séquences de cartes comme :

$$\boxed{?}\boxed{?}\boxed{\spadesuit}\boxed{\heartsuit}\boxed{\spadesuit}\boxed{\heartsuit}\dots$$

On définit *l'encodage* d'un booléen de la manière suivante :

- 0 est encodé par $\boxed{\spadesuit}\boxed{\heartsuit}$;
- 1 est encodé par $\boxed{\heartsuit}\boxed{\spadesuit}$.

On généralise l'encodage à des n-uplets de booléens en plaçant les encodages côte à côte. On écrit par exemple $\underbrace{\boxed{?}\boxed{?}}_{b}\boxed{\spadesuit}\boxed{\heartsuit}\boxed{\spadesuit}\boxed{\heartsuit}$ pour une séquence de 6 cartes encodant $(b, 0, 0)$.

Soit $f : \{0,1\}^n \to \{0,1\}$ une fonction booléenne. Un *protocole de calcul* à cartes pour la fonction f est un algorithme qui :

1. Prend en entrée une séquence de $m = 2n + m_b + m_n$ cartes de la forme :

$$\underbrace{\boxed{?}\boxed{?}}_{b_0}\underbrace{\boxed{?}\boxed{?}}_{b_1}\dots\underbrace{\boxed{?}\boxed{?}}_{b_{n-1}}\underbrace{\boxed{\heartsuit}\boxed{\heartsuit}\dots\boxed{\heartsuit}}_{m_b}\underbrace{\boxed{\spadesuit}\boxed{\spadesuit}\dots\boxed{\spadesuit}}_{m_n},$$

   encodant un vecteur booléen secret $(b_0, \dots, b_{n-1})$ ainsi qu'un nombre fixé de cartes additionnelles face visible.

2. Peut appliquer les opérations suivantes :
   - **Permute**(π) : permutation des cartes selon une permutation fixe π de $\{0, \dots, m-1\}$:

$$(\alpha_0, \dots, \alpha_{m-1}) \to (\alpha_{\pi(0)}, \dots, \alpha_{\pi(m-1)}).$$

   - **Coupe**$(i, 2j)$: dans la sous-séquence $(\alpha_i, \dots, \alpha_{i+2j-1})$, on applique une *coupe* aléatoire :

$$(\alpha_i, \dots, \alpha_{i+j-1}, \alpha_{i+j}, \dots, \alpha_{i+2j-1}) \to \begin{cases} (\alpha_i, \dots, \alpha_{i+j-1}, \alpha_{i+j}, \dots, \alpha_{i+2j-1}) \\ (\alpha_{i+j}, \dots, \alpha_{i+2j-1}, \alpha_i, \dots, \alpha_{i+j-1}) \end{cases}$$

     où chaque cas arrive avec probabilité 1/2. En particulier, en appliquant une coupe sur des cartes face cachée, il est impossible de dire quel cas s'est produit.
   - **Retourne**(i) : retourne la carte en position i, qui passe donc de recto à verso ou inversement.

3. Est tel qu'à la fin du protocole, les deux premières cartes de la séquence, face verso, encodent $f(b_0, \dots, b_{n-1})$. Les autres cartes peuvent être dans un état quelconque.

Les actions effectuées peuvent dépendre des symboles observés sur les cartes retournées. De plus, on exige que le protocole ne révèle aucune information sur les entrées $b_0, \dots, b_{n-1}$. On pourra justifier cette propriété de manière informelle.

Question 1. Donner un protocole calculant la fonction NOT sans révéler la valeur de son entrée :

$$\underbrace{\boxed{?}\boxed{?}}_{b} \to \underbrace{\boxed{?}\boxed{?}}_{\neg b}.$$

Question 2. Sur l'entrée encodant $\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\underbrace{\boxed{\spadesuit}\boxed{\heartsuit}}_{0}$, on effectue les opérations suivantes :

1. Retourner les deux cartes visibles face verso.
2. Appliquer la permutation
$$(1,2,3,4,5,6) \to (1,3,5,2,4,6).$$
3. Appliquer une coupe aléatoire (avec probabilité 1/2) :
$$(1,2,3,4,5,6) \to \begin{cases} (1,2,3,4,5,6) \\ (4,5,6,1,2,3) \end{cases}$$
4. Appliquer ensuite la permutation
$$(1,2,3,4,5,6) \to (1,4,2,5,3,6).$$

Montrer que la séquence de cartes obtenue est un encodage de : $(a \oplus r, b \oplus r, r)$, où $\oplus$ est le OU exclusif, et r vaut 0 ou 1 avec probabilité 1/2.

Question 3. En déduire :

1. Un protocole pour l'opération COPY :
$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{\spadesuit}\boxed{\heartsuit}}_{0}\underbrace{\boxed{\spadesuit}\boxed{\heartsuit}}_{0} \to \underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{\spadesuit}\boxed{\heartsuit}}_{0}.$$
2. Un protocole pour l'opération XOR.
$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\underbrace{\boxed{\spadesuit}\boxed{\heartsuit}}_{0} \to \underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{a\oplus b}\underbrace{\boxed{\spadesuit}\boxed{\heartsuit}}_{0}.$$

Question 4. Sur une entrée $\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\spadesuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b}$ encodant deux bits a et b et deux cartes supplémentaires $\boxed{\spadesuit}\boxed{\heartsuit}$, on effectue les opérations suivantes :

1. Retourner les deux cartes visibles face verso.
2. Appliquer la permutation
$$(1,2,3,4,5,6) \to (1,3,4,2,5,6).$$
3. Appliquer une coupe aléatoire (avec probabilité 1/2) :
$$(1,2,3,4,5,6) \to \begin{cases} (1,2,3,4,5,6) \\ (4,5,6,1,2,3) \end{cases}$$
4. Appliquer ensuite la permutation
$$(1,2,3,4,5,6) \to (1,4,2,3,5,6).$$

Compléter ce protocole et montrer qu'il permet de renvoyer $a \wedge b$.

Question 5. Compléter le protocole précédent pour qu'il renvoie $\boxed{\spadesuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b}\underbrace{\boxed{?}\boxed{?}}_{a\wedge b}$.

Question 6. Montrer que pour toute fonction booléenne $f : \{0,1\}^n \to \{0,1\}$ il existe un protocole à cartes qui la calcule permettant de la calculer. De combien de cartes auxiliaires a-t-on besoin ?

Question 7. Donner un protocole de « *random flip* » qui sur une entrée $(b_0, \dots, b_{n-1})$ (face verso) renvoie $(b_0 \oplus r, \dots, b_{n-1} \oplus r)$ où r est une valeur aléatoire uniforme (face verso). En déduire un protocole à $2n$ cartes calculant la fonction « égalité » :

$$f(b_0, \dots, b_{n-1}) = \begin{cases} 1 & \text{si } b_0 = \cdots = b_{n-1} \\ 0 & \text{sinon} \end{cases}$$

Une fonction booléenne est dite *symétrique* si $f(\pi(b_0, \dots, b_{n-1})) = f(b_0, \dots, b_{n-1})$ pour toute permutation π de $\{0, \dots, n-1\}$.

Question 8. Donner un protocole calculant sur l'entrée $(b_0, \dots, b_{n-1})$ un encodage de $\sum b_i$.
En déduire un protocole pour les fonctions symétriques avec seulement deux cartes additionnelles pour $n \geqslant 4$.