

TP X2

Masques binaires

Instructions

Les questions marquées d'astérisques (* ou **) peuvent être plus difficiles ou demander plus de temps.

1 Introduction

Les **masques binaires** (*bitmasks* en Anglais) offrent une représentation efficace des listes de Booléens.

À titre d'exemple, nous traitons la représentation d'ensembles finis d'entiers naturels.

2 Ensembles finis d'entiers naturels

Nous commençons par une représentation des ensembles finis d'entiers naturels par des listes de Booléens. Nous verrons une représentation avec des masques binaires en §3.

Une liste de Booléens \mathbf{s} représente l'ensemble d'entiers

$$\{n \mid 0 \leq n < \text{len}(\mathbf{s}) \text{ et } \mathbf{s}[n] = \text{True}\}$$

En particulier, la liste vide $[]$ représente l'ensemble vide \emptyset , et la liste $[\text{True}]$ représente le singleton $\{0\}$.

Question 1.

Q.1

(1) Donner l'ensemble représenté par la liste $[\text{True}, \text{False}, \text{True}, \text{True}, \text{False}, \text{True}]$.

(2) Donner une liste représentant l'ensemble $\{2, 4, 5, 6\}$.

Question 2. Écrire une fonction Python singleton qui prend en argument un entier n et qui renvoie une liste représentant l'ensemble singleton $\{n\}$.

Q.2

Question 3. Écrire une fonction Python appartient(\mathbf{s}, n) qui renvoie True si l'entier n appartient à l'ensemble représenté par la liste de Booléens \mathbf{s} , et qui renvoie False sinon.

Q.3

3 Représentation binaire d'ensembles finis d'entiers naturels

Nous allons maintenant représenter une liste de Booléens $[b_0, \dots, b_{n-1}]$ par l'entier

$$(a_{n-1} \dots a_0)_2$$

où $a_i = 1$ si $b_i = \text{True}$ et $a_i = 0$ si $b_i = \text{False}$.

Attention à l'ordre inversé entre b_0, \dots, b_{n-1} et $a_{n-1} \dots a_0$. Par exemple, l'entier $(1101)_2 = 13$ représente la liste $[\text{True}, \text{False}, \text{True}, \text{True}]$.

Question 4. Donner l'entier représentant la liste $[\text{False}, \text{False}, \text{True}, \text{True}, \text{False}, \text{True}]$.

Q.4

Question 5. Écrire une fonction Python encode(\mathbf{s}) qui renvoie l'entier représentant la liste de Booléens \mathbf{s} .

Q.5

Question 6. Écrire une fonction Python `decode(c)` qui renvoie une liste de Booléens représentée par l'entier c .

Q.6

La représentation des listes de Booléens par des entiers offre, lorsqu'elle est combinée avec la représentation vue en §2, une représentation des ensembles finis d'entiers par des entiers. Dans cette représentation, l'ensemble vide \emptyset est représenté par l'entier 0, et le singleton $\{0\}$ est représenté par l'entier $1 = 1_2$.

Question 7. Donner l'entier représentant l'ensemble $\{2, 4, 5, 6\}$.

Q.7

Pour manipuler efficacement la représentation binaire, il faut utiliser les opérations bit-à-bit vues au §7 du TP1.¹

Question 8. Écrire une fonction Python `code_singleton` qui prend en argument un entier n et qui renvoie l'entier représentant l'ensemble singleton $\{n\}$.

Q.8

Question 9. Écrire une fonction Python `code_union(c,d)` telle que si les entiers c et d représentent les ensembles S et T respectivement, alors `code_union(c,d)` renvoie l'entier représentant l'ensemble $S \cup T$.

Q.9

Question 10. Écrire une fonction Python `code_intersection(c,d)` telle que si les entiers c et d représentent les ensembles S et T respectivement, alors `code_intersection(c,d)` renvoie l'entier représentant l'ensemble $S \cap T$.

Q.10

Question 11. Écrire une fonction Python `code_appartient(c,n)` qui renvoie `True` si l'entier n appartient à l'ensemble représenté par l'entier c , et qui renvoie `False` sinon.

Q.11

4 Exercices supplémentaires

À titre de comparaison avec les fonctions du §3, il est intéressant de voir comment implémenter les opérations ensemblistes \cup, \cap directement sur les listes représentant des ensembles comme en §2, sans passer par la représentation binaire.

Question 12. Écrire une fonction Python `union(s,t)` telle que si s et t sont des listes de Booléens qui représentent les ensembles S et T respectivement, alors `union(s,t)` renvoie une liste représentant l'ensemble $S \cup T$.

Q.12

Question 13. Écrire une fonction Python `intersection(s,t)` telle que si s et t sont des listes de Booléens qui représentent les ensembles S et T respectivement, alors `intersection(s,t)` renvoie une liste représentant l'ensemble $S \cap T$.

Q.13

Les questions suivantes utilisent des opérations bit-à-bit supplémentaires.²

Question 14. Pour chacune des deux représentations, écrire une fonction Python qui prend en arguments les représentations de deux ensembles S et T , et qui renvoie la représentation de l'ensemble $S \setminus T$.

Q.14

Question 15. Pour chacune des deux représentations, écrire une fonction Python qui prend en arguments les représentations de deux ensembles S et T , et qui renvoie la représentation de la différence symétrique de S et T , c'est-à-dire de l'ensemble $(S \setminus T) \cup (T \setminus S)$.

Q.15

1. Le TP1 est disponible à <https://perso.ens-lyon.fr/colin.riba/teaching/cpes/tp/tp01.pdf>.

2. Voir <https://wiki.python.org/moin/BitwiseOperators>.