

# Planche 1 type CCINP

## 1 PARTIE A

---

On va montrer que 3SAT peut se réduire au problème MAX2SAT dont la formulation est la suivante : Etant donné une formule  $\varphi$  sous forme 2SAT et un entier  $k$  est-il possible de satisfaire simultanément au moins  $k$  clauses de  $\varphi$ .

1. On fixe  $l_1, l_2$  et  $l_3$  trois littéraux et une variable  $x$  indépendante des trois littéraux. On note  $\bar{l}_1$  la négation de  $l_1$ . On considère alors le groupe de 10 clauses (on remarquera le rôle symétrique des variables  $l_i$ ) suivant :

$$l_1 \wedge l_2 \wedge l_3 \wedge x \wedge (\bar{l}_1 \vee \bar{l}_2) \wedge (\bar{l}_2 \vee \bar{l}_3) \wedge (\bar{l}_1 \vee \bar{l}_3) \wedge (l_1 \vee \neg x) \wedge (l_2 \vee \neg x) \wedge (l_3 \vee \neg x)$$

- (a) Montrer que si  $l_1 \vee l_2 \vee l_3$  est valide alors on peut donner à  $x$  une valeur de vérité telle que 7 des 10 clauses sont satisfaites, mais pas plus.
  - (b) Montrer que si  $l_1 \vee l_2 \vee l_3$  n'est pas valide alors on ne peut pas satisfaire plus de 6 clauses en même temps.
2. Soit  $\varphi$  une formule avec  $c$  clauses sous forme 3SAT, définir une formule  $\varphi'$  de taille polynomiale en la taille de  $\varphi$  et  $k$  un entier tels que  $\varphi$  est satisfiable si et seulement s'il existe une valuation pour  $\varphi'$  satisfaisant au moins  $k$  clauses.

## 2 PARTIE B

---

Connaissez-vous le jeu du cadavre exquis ? Vous pouvez décider d'y jouer seul, avec votre ordinateur. C'est ce que vous allez faire maintenant.

La première approche, qui consiste à choisir des mots au hasard les uns à la suite des autres, ne produit pas des phrases ressemblant à du français. On va commencer par se fixer une structure grammaticalement correcte.

Dans cette partie, on cherche à produire au hasard une liste du genre [Nom; Verbe; Nom; Adjectif] décrivant la nature des mots de la phrase. Il n'y aura plus, ensuite, qu'à remplacer chaque nature par un mot convenable pour obtenir une phrase grammaticalement correcte (ou presque). Pour prendre en compte les accords entre les différents mots (ici le dernier Nom et son Adjectif doivent être accordés), on stocke aussi les informations de genre et de nombre.

Voici le type `caml` qui sera utilisé pour représenter les natures :

```
type genre = Masc | Fem
type nombre = Sing | Plur
type nature =
  | Nom of (genre * nombre)
  | Verbe of nombre
  | Adjectif of (genre * nombre)
```

▷ **Question 1.** Écrire des fonctions `choisir_nombre : unit -> nombre` et `choisir_genre : unit -> genre` qui choisissent un nombre et un genre au hasard. On pourra utiliser la fonction `Randdom.bool()` choisissant au hasard un booléen. ◀

▷ **Question 2.** Écrire une fonction `gn : (genre * nombre) -> nature list` générant un groupe nominal, c'est-à-dire un nom suivi avec une probabilité de 0.3 par un adjectif, respectant les accords (genre et nombre) donnés en paramètres. ◀

▷ **Question 3.** Écrire une fonction `structure_phrase : unit -> nature list` qui génère une phrase complète selon le schéma "Sujet - Verbe - COD". Le sujet et le COD seront des groupes nominaux. ◀

Pour passer d'une structure à une phrase complète, il faut remplacer chaque nature par un mot qui convient. On utilisera dans cet exercice l'approche la plus simple possible : une table qui associe à chaque nature un tableau de mots (par exemple `Nom (Masc, Sing) -> [l "eleve"; "retardataire"]`).

Le code compagnon vous fournit un exemple d'une telle table.

▷ **Question 4.** Écrire une fonction `remplir_mot : nature -> string` qui choisit un mot de la bonne nature dans la table.

Adapter `remplir_mot` pour ajouter devant les noms un article convenablement accordé, en utilisant une fonction auxiliaire `choisir_article : string -> (genre * nombre) -> string` qui prend en paramètre le nom et son accord). ◀

▷ **Question 5.** Écrire une fonction `phrase_hasard () : unit -> unit` qui génère une phrase aléatoire (en utilisant `structure_phrase` et `remplir_mot`) et l'affiche. ◀

▷ **Question 6.** Adapter pour pouvoir éventuellement (avec probabilité 0,2 par exemple) ajouter le mot "avec" devant le COD. ◀