

Exercice d'application liste : le jeu des deux tribus

On considère une population constituée de 2 tribus antagonistes : les + et les -. Cette population peut être représentée, à chaque instant, par une liste de symboles, chacun des symboles étant un + ou un -. Par exemple, la population (+ + + + -) contient 4 individus de la tribu + et 1 individu de la tribu -. Le but du « jeu » est d'obtenir une population avec une tribu unique : tous les + ont été anéantis, ou bien tous les - ont été anéantis. Pour jouer, on dispose des 4 règles suivantes :

- | | |
|------------------------------------|---------------------------------------|
| 1) (...+ - ...) → (... - + ...) | permutation d'un + et d'un - contigus |
| 2) (... - + ...) → (... + - ...) | permutation d'un - et d'un + contigus |
| 3) (... - + - ...) → (... - - ...) | suppression d'un + encadré par deux - |
| 4) (... + - + ...) → (... + + ...) | suppression d'un - encadré par deux + |

1- Définir la fonction **permutel** qui s'applique à une population et effectue la première permutation possible. Le résultat retourné est la population modifiée. Si aucune permutation n'est possible, la fonction retourne une liste vide.

2- Définir la fonction **permute** qui s'applique à une population et effectue toutes les permutations possibles en appliquant les règles 1 et 2 ci-dessus. Le résultat retourné est la liste des populations issues de ces permutations. Si aucune permutation n'est possible, la fonction retourne une liste vide.

Ex : (*permute* '(+ - - -)) → ((- + + -) (+ + - -) (+ - + -))

3- Définir la fonction **mange1** qui s'applique à une population et supprime le premier individu qui peut être supprimé. Le résultat retourné est la population modifiée. Si aucune suppression n'est possible, la fonction retourne une liste vide.

4- Définir la fonction **mange** qui s'applique à une population et effectue toutes les suppressions possibles dans cette population en appliquant les règles 3 et 4 ci-dessus. Le résultat retourné est la liste des populations issues de ces suppressions. Si aucune suppression n'est possible, la fonction retourne une liste vide.

Ex : (*mange* '(+ - - -)) → ((+ + -) (+ - -))

5- Définir la fonction **suivantes** qui s'applique à une population et retourne la liste des populations obtenues par application des règles du jeu. Si aucune règle n'est applicable la fonction **suivantes** retourne une liste vide.

Ex : (*suivantes* '(+ - +)) → ((- + +) (+ + -) (+ +))

6- Définir le prédicat **une-solution?** qui s'applique à une population. S'il existe une solution du jeu pour cette population, le prédicat affiche la population solution et retourne #t. Dans le cas contraire, le prédicat retourne #f. Pour faciliter l'écriture de ce prédicat, on pourra définir le prédicat *ou-solution* qui s'applique à un prédicat *p* et à une liste d'éléments et qui est défini de la manière suivante :

(*ou-solution* *p* '(*e1 e2...en*)) ⇔ (or (*p e1*) (*p e2*) ... (*p en*))