

Spécification de l'implémentation du ruban (SGR A)

1 **Squelette de la classe Java TapeA_B implémentant l'interface Java Tape**

```
public class TapeA_B implements Tape {
    private Cell readhead;

    private int repOkNb = -1;
    private boolean nullNextEncountered;
    private boolean nullPreviousEncountered;

    public TapeA_B(String init) throws Exception {
    }

    public boolean repOk() {
        return false;
    }

    public boolean isSymbol(char s) throws Exception {
        return false;
    }

    public void putSymbol(char s) throws Exception {
    }

    public void leftMove() {
    }

    public void rightMove() {
    }

    public String toString() {
        return null;
    }

    private class Cell {
        char content;
        Cell next = null;
        Cell previous = null;
        int checkedFor = -1;

        public Cell() {
            content = (char) 32;
        }
    }
}
```

2 Conventions de représentation

Un ruban de machine de Turing est composé d'une suite infinie de cases. Toutes ces cases du ruban sont blanches, sauf un nombre fini d'entre elles. On peut donc diviser conceptuellement ce ruban en trois parties. En parcourant le ruban de gauche à droite, on détermine respectivement les 3 zones A, B et C.

Les parties A et C sont composées d'une infinité de cases blanches, tandis que la partie centrale B est composée d'un nombre fini de cases, contenant uniquement des symboles appartenant à l'alphabet Gamma défini ci-dessous, non nécessairement blancs.

A tout instant, la zone B doit être la plus petite possible. Ainsi, la délimitation entre les 3 parties ne peut jamais être ambiguë. En particulier, à une exception près, cela signifie que les cases situées aux extrémités intérieures de la zone B, ne peuvent être blanches. On considère néanmoins que la zone B doit être constituée au minimum d'une case, même si celle-ci est blanche, auquel cas on peut considérer que ses bords sont blancs.

Nous représentons ce ruban par une liste doublement chaînée. La zone du ruban représentée par cette liste est une zone B'. Cette zone B' comprend la zone B dans son entièreté, et peut être étendue de façon infinie dans les deux sens par des cases blanches, permettant d'accompagner le déplacement de la tête de lecture. A tout instant, cette zone B' doit être la plus petite possible.

Chaque élément de la liste chaînée contient : un lien vers l'élément suivant et un lien vers l'élément précédent dans la liste ; le symbole contenu dans la case correspondante du ruban ; une variable qui sera utilisée comme marqueur pour permettre une vérification efficace du respect de l'invariant de classe du ruban.

Un élément de cette liste doublement chaînée ayant un pointeur next ou previous null représente une case située à une des extrémités de la zone B'. Si le symbole contenu dans cette cellule correspond au caractère blanc, cette cellule est alors obligatoirement celle située sous la tête de lecture.

Ce choix de représentation du ruban par une liste doublement chaînée permet de pousser la limite du nombre de cases composant ce ruban à la mémoire disponible sur l'ordinateur exécutant le programme.

Le ruban comprend quatre variables d'instances : Cell readhead, pointant vers la cellule de la liste chaînée correspondant à la case du ruban située sous la tête de lecture ; int repOkNb, utilisée en coordination avec le marqueur int checkedFor contenu dans les cellules de la liste chaînée, pour permettre une vérification efficace de l'invariant ; boolean nullNextEncountered et boolean nullPreviousEncountered également utilisées pour la vérification du respect de l'invariant, et permettre de s'assurer que toutes les cases non blanches du ruban peuvent être atteintes lors du parcours du ruban. Les trois variables d'instance repOkNb, nullNextEncountered et nullPreviousEncountered seront utilisées pour l'implémentation de la méthode repOk().

L'alphabet Gamma se compose des caractères définis par le code ASCII, représentés en décimal par les nombres compris entre 32 et 126 inclus.

3 Invariant de représentation

- les symboles représentés sur les cellules de la liste chaînée appartiennent tous à l'alphabet Gamma.
- la liste chaînée représentant le ruban ne peut boucler. Une cellule ne peut pointer sur elle-même, ni même sur une cellule déjà référencée par un pointeur du même type (next ou previous). En d'autres mots, le parcours de cette liste dans son entièreté assure un passage unique sur chacune des cases de la liste.
- une et une seule cellule de la liste chaînée possède un pointeur vers la cellule précédente (previous) de valeur null.
- une et une seule cellule de la liste chaînée possède un pointeur vers la cellule suivante (next) de valeur null.
- la tête de lecture pointe vers une case appartenant au ruban. Sa valeur ne peut être null.
- la zone B est la plus petite possible.
- la zone B' est la plus petite possible.