

1. Représentation des intervalles

En période 5, vous avez réalisé un petit outil d'analyse statique qui calculait dans le treillis des signes. Lors de cette période, nous étendons l'outil pour qu'il calcule maintenant dans le treillis des intervalles.

Rappel : le treillis des intervalles est défini comme suit :

$$\mathbb{I} = \{[a, b] : a, b \in \mathbb{Z}\} \cup \{[a, +\infty[: a \in \mathbb{Z}\} \cup \{]-\infty, b] : b \in \mathbb{Z}\} \cup \{]-\infty, +\infty[\}$$

Il est muni de l'ordre *plus précis que* qui représente l'inclusion des intervalles : pour toute paire d'intervalles (a, b) et (c, d) , dans \mathbb{I} (les parenthèses peuvent représenter des crochets ouvrants ou fermants selon le besoin)

$$(a, b) \leq (c, d) \text{ ssi } c \leq a \text{ et } b \leq d$$

où l'ordre sur les entiers est étendu naturellement à $\mathbb{Z} \cup \{+\infty, -\infty\}$.

Nous avons montré en cours que \mathbb{I} muni de cet ordre est bien un treillis complet.

Le but du premier TP est d'implémenter des classes pour représenter les intervalles.

Code fourni

Nous vous conseillons de repartir du code fourni pour cette deuxième période (*cf.* Chamilo). Par rapport à la version précédente (et peut-être à la vôtre), il contient en plus :

- l'algorithme qui déplie les conditions de façon à n'avoir que des conditions simples,
- le merge des transitions en sortie de if et de while,
- une sortie sous forme de tableaux des résultats de l'analyseur.

Le squelette des classes à implémenter est dans `IntervalLattice.java`.

Bornes d'un intervalle

La classe `Bound` sert à représenter les bornes des intervalles. Autrement dit, un objet de cette classe est un élément de $\mathbb{Z} \cup \{+\infty, -\infty\}$: il peut valoir soit $+\infty$, soit $-\infty$, soit un entier relatif.

Choisissez une représentation de la classe `Bound` puis implémentez :

- des comparateurs (`equals`, `≤`, `<`, `≥`, `>`),
- des tests (égal à 0, est positif, est négatif, est positif ou nul, est négatif ou nul) ;
- des opérations binaires (+, −, ×, min, max)
- une méthode `toString`.

Testez !

Intervalles

La classe `Interval` sert à représenter les intervalles du treillis des intervalles à l'aide de deux bornes (de classe `Bound`). Elle doit définir et donner accès à la plus petite et à la plus grande valeurs du treillis (appelées BOT et TOP dans le code).

Choisissez une représentation pour BOT et TOP. Implémentez ensuite la classe `Interval` avec :

- le / les constructeurs nécessaires,
- les accesseurs (getters),
- les méthodes `equals` et `toString`.

Testez !