

# Compte rendu des réunions

## Sweep as a Generic Pruning Technique Applied to the Non-Overlapping Rectangles Constraint

### Réunion 3 :

Description de nos deux fonctions principales :

`findMinimum(Constraint rectangle)` qui permet de trouver la position minimale en  $x$  d'un rectangle par rapport aux contraintes existantes.

`nonOverLapLeft()` prend l'ensemble des contraintes du domaine et fait un `findMinimum` pour trouver la position minimale en  $x$  et met à jour la borne de la contrainte.

Maintenant que nous arrivons à mettre à jour la borne minimale en  $x$  pour toutes les contraintes du domaine, nous désirons mettre à jour la borne maximale en  $x$ , minimale en  $y$  et maximale en  $y$ .

Nous allons donc créer une méthode `nonOverLap` qui sera capable de recalculer tous ces types de bornes. Pour savoir dans quel sens effectuer le balayage, la méthode prendra deux paramètres : `calculateAbscisse` (true pour  $x$  et false pour  $y$ ) et `calculateMin` (true pour min et false pour max).

La prochaine étape est d'effectuer un `nonOverLap` dans chaque sens. Il faudra vérifier à chaque balayage si une borne a été mise à jour, dans ce cas il faudra recommencer le balayage à zéro (dans tous les sens de nouveau) jusqu'à ne plus mettre à jour.