

---

# Rapport de TP : un planning pour des infirmières

Logique des systèmes déductifs

*Benoit Valiron*

---

Mathis Benyahia

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Mode d'emploi . . . . .	1
<b>2</b>	<b>Choix dans la Structure des Données</b>	<b>2</b>
<b>3</b>	<b>Explication des Différentes Contraintes</b>	<b>3</b>
3.1	Contrainte 0 : Week-end non sécable . . . . .	3
3.2	Contrainte 1 : Enchaînement des heures de travail . . . . .	3
3.3	Contrainte 2 : Jours de travail consécutifs . . . . .	3
3.4	Contrainte 3 : Nombre d'infirmières par période . . . . .	3
3.5	Contrainte 4 : Nombre total de demi-journées travaillées . . . . .	3
<b>4</b>	<b>Chapitre4</b>	<b>4</b>
4.1	Format du Résultat Obtenu . . . . .	4
4.2	Ce qui Fonctionne . . . . .	4
4.3	Ce qui ne Fonctionne Pas . . . . .	4
4.4	Solutions Proposées par l'optimiseur de z3 . . . . .	4
4.4.1	Planning des vacataires . . . . .	4
4.4.2	Planning de l'infirmière 1 . . . . .	4
4.4.3	Planning de l'infirmière 2 . . . . .	5
4.4.4	Planning de l'infirmière 3 . . . . .	5
4.4.5	Planning de l'infirmière 4 . . . . .	5

# Chapitre 1

## Introduction

Le but de ce TP est d'appréhender l'outil z3 via son API python. Pour cela nous allons résoudre un problème courant. La construction d'un planning à partir d'un certain nombre de contraintes. Dans notre cas il s'agit d'un hôpital comportant 4 infirmières. L'hôpital peut faire appel à des vacataires mais il faut en appeler le moins possible.

### 1.1 Mode d'emploi

Il y a associé à ce rapport un fichier *planning.py*. Ce fichier produit une solution pour les 4 infirmières ainsi que les vacataires. On peut observer la solution soit directement dans le terminal soit dans les 4 fichiers .csv créés dans le même répertoire que le fichier python.

Il suffit de lancer le fichier python pour obtenir les planning.

## Chapitre 2

# Choix dans la Structure des Données

Pour modéliser le problème de planification, nous avons utilisé les types énumérés suivants :

- **Week** : Représente les semaines (W1, W2, W3, W4).
- **Day** : Représente les jours de la semaine (Monday, Tuesday, Wednesday, Thursday, Friday, We).
- **Hour** : Représente les heures de la journée (AM, PM).
- **Nurse** : Représente les infirmières (N1, N2, N3, N4).
- **Status** : Représente le statut de chaque infirmière (Work, Rest).

Nous avons également défini des listes pour les jours de la semaine, les semaines, les infirmières et les heures de la journée pour faciliter la manipulation des données.

Enfin on construit le planning à partir de deux éléments :

- Une fonction *Planning* qui prend en entrée une semaine *Week*, un jour *Day*, une heure *Hour* ainsi qu'une infirmière *Nurse* et retourne un statut *Status* pour indiquer si sur un créneau l'infirmière travaille.
- Un ensemble d'entiers *Int* qui représentent le nombre de vacataires requis pour chaque créneau.

Du côté de z3, on utilise plutôt l'optimiseur plutôt que le solveur car on veut d'une part trouver un planning qui respecte les contraintes mais en plus, on veut minimiser le recours aux vacataires (c'est à dire à la somme du nombre de vacataires par créneau nécessaires sur 4 semaines)

Enfin notre code repose sur quelques fonctions utilitaires bien utiles :

- *print\_planning(planning, infirmiere, save)* : qui prend le planning brut et en sort une représentation dans la console et permet de l'enregistrer
- *print\_vacataires(planning\_vacataires, save)* : qui prend le planning brut des besoins en vacataires et en sort une représentation dans la console et permet de l'enregistrer.
- *calc\_creneau\_suivant(week, day, hour)* : retourne le créneau suivant celui qui est donné en paramètre.
- *calc\_creneau\_precedent(week, day, hour)* : retourne le créneau precedent celui qui est donné en paramètre.

Enfin nous avons pris en compte la dimension cyclique du planning. Donc les conditions sont respectées même lorsqu'on reprend la W1 après la W4. Le respect de la cyclicité repose sur sa prise en compte dans les fonctions de calcul de créneau précédent et suivant. Cela simplifie par ailleurs grandement les contraintes car nous n'avons pas de cas aux extrémités du planning à traiter.

## Chapitre 3

# Explication des Différentes Contraintes

Nous avons implémenté plusieurs contraintes comme demandé dans l'énoncé du TP. Ces contraintes semblent principalement être issues de contraintes réglementaires.

### 3.1 Contrainte 0 : Week-end non sécable

Le week-end (We) ne peut pas être divisé. Si une infirmière travaille le samedi matin (AM), elle doit également travailler le samedi après-midi (PM), et vice versa. De plus le week-end compte comme 4 demi-journées (deux jours même si réuni en 1 dans nos énumérations).

### 3.2 Contrainte 1 : Enchaînement des heures de travail

Un enchaînement d'un soir et d'un matin est impossible.  
De plus, une journée complète de travail implique une journée complète de repos avant et après.

### 3.3 Contrainte 2 : Jours de travail consécutifs

Pas plus de 6 jours de travail consécutifs sans journée de repos. Cela signifie qu'après 6 jours de travail, une infirmière doit avoir une journée de repos. Pour nous aider à écrire cette contrainte, nous avons utilisé une fonction intermédiaire qui calcule le nombre de jours travaillés sur les 6 derniers jours avec la sémantique de z3.

### 3.4 Contrainte 3 : Nombre d'infirmières par période

Il faut 3 infirmières le matin en semaine, 2 infirmières le soir en semaine et 2 infirmières le week-end. Cela garantit une couverture adéquate à tout moment.

### 3.5 Contrainte 4 : Nombre total de demi-journées travaillées

Chaque infirmière doit travailler exactement 20 demi-journées sur 4 semaines. Cet élément est clé dans notre calcul du nombre de vacataires nécessaires. Si on fait le calcul sur 4 semaines, il y a 132 demi-journées à affecter. Or chaque infirmière peut travailler 20 demi-journées dans le mois. Ainsi, nous aurons besoin au minimum de 52 demi-journées de vacataires dans le mois. On pourra ainsi vérifier que dans les planning produit on atteint ce minimum théorique de vacataires. Si on voulait complètement se passer de vacataires, il faudrait en théorie au minimum 7 infirmières.

## Chapitre 4

# Chapitre4

### 4.1 Format du Résultat Obtenu

Le résultat est affiché sous forme de tableau pour chaque infirmière, indiquant les périodes de travail (W) et de repos (R) pour chaque jour de la semaine et chaque semaine. De plus, un tableau séparé montre le nombre de vacataires nécessaires pour chaque période.

### 4.2 Ce qui Fonctionne

Le code utilise la bibliothèque Z3 pour résoudre le problème de planification. Si une solution est trouvée, elle est affichée et enregistrée dans des fichiers CSV. Le code gère également les cas où aucune solution n'est trouvée en affichant "unsat".

### 4.3 Ce qui ne Fonctionne Pas

Actuellement, le code ne fournit pas de commentaires détaillés expliquant chaque partie du code. De plus, il n'y a pas de mécanisme pour gérer les cas où aucune solution n'est trouvée, à part afficher "unsat".

### 4.4 Solutions Proposées par l'optimiseur de z3

#### 4.4.1 Planning des vacataires

Le premier chiffre est le nombre nécessaire pour le matin et le second pour l'après-midi.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	We
Week 1	2/2	1/0	1/1	2/0	2/2	0/0
Week 2	3/0	2/0	2/1	0/1	3/1	0/0
Week 3	1/1	3/0	2/2	2/1	1/1	0/0
Week 4	1/2	1/1	0/2	0/2	2/1	0/0

On observe qu'on a recours à 52 demi-journées de vacataires. Comme on l'a expliqué plus tôt c'est bien le minimum pour cette configuration.

#### 4.4.2 Planning de l'infirmière 1

La première lettre est le statut du matin et la seconde le statut de l'après-midi. *R* pour Rest et *W* pour Work.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	We
Week 1	R/R	W/R	W/R	W/R	R/R	W/W
Week 2	R/R	W/R	R/R	W/W	R/R	R/R
Week 3	W/W	R/R	W/R	W/R	W/R	R/R
Week 4	W/R	W/R	W/R	W/R	W/R	R/R

#### 4.4.3 Planning de l'infirmière 2

La première lettre est le statut du matin et la seconde le statut de l'après-midi. *R* pour Rest et *W* pour Work.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	We
Week 1	R/R	R/W	R/W	R/W	R/R	R/R
Week 2	R/W	R/W	R/W	R/R	R/R	W/W
Week 3	R/R	R/R	R/R	R/W	R/R	W/W
Week 4	R/R	R/W	R/R	R/R	R/R	W/W

#### 4.4.4 Planning de l'infirmière 3

La première lettre est le statut du matin et la seconde le statut de l'après-midi. *R* pour Rest et *W* pour Work.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	We
Week 1	W/R	W/R	W/R	R/W	R/R	W/W
Week 2	R/R	R/R	W/R	W/R	R/W	R/R
Week 3	W/R	R/W	R/R	R/R	W/W	R/R
Week 4	W/R	W/R	W/R	W/R	R/W	R/R

#### 4.4.5 Planning de l'infirmière 4

La première lettre est le statut du matin et la seconde le statut de l'après-midi. *R* pour Rest et *W* pour Work.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	We
Week 1	R/R	R/W	R/R	R/R	W/R	R/R
Week 2	R/W	R/W	R/R	W/R	R/R	W/W
Week 3	R/R	R/W	R/R	R/R	R/R	W/W
Week 4	R/R	R/R	W/R	W/R	R/R	W/W