

CSP Exercices

Pour ces exercices, nous allons utiliser une bibliothèque logicielle pour la résolution de contraintes à partir du langage de programmation Python. Réalisez les étapes suivantes **avant le cours** si vous voulez faire les exercices sur votre propre ordinateur.

- A) Si ce n'est pas déjà fait, allez sur python.org, onglet Downloads et téléchargez la dernière version de Python. Faites exécuter le programme d'installation (.exe), Install Now et choisissez de mettre Python dans le PATH.
- B) Sur la ligne de commande (menu démarrer, mettre cmd dans la case), faites `pip3 install jupyterlab`¹.
- C) Pour installer la bibliothèque Z3, développée chez Microsoft Research et qui permet de résoudre des contraintes, dans plusieurs langages, par réduction au problème SAT, il faut procéder comme suit. Créez tout d'abord un répertoire de nom *DIC9150* dans votre répertoire personnel. Allez sur github.com/Z3Prover/z3/releases, téléchargez la **Latest release** (z3-XXX-x64-win.zip pour 64 bits et z3-XXX-x86-win.zip pour Python 32 bits). Double-cliquez sur l'archive .zip, faites faire l'extraction et mettez votre répertoire *DIC9150* comme destination.

Exercices

1. Avec le bouton de droite de la souris, sélectionnez le fichier *mspEx.ipynb* du site web du cours et enregistrez le dans le répertoire *z3-XXX-.../bin/python* qui se trouve maintenant dans votre répertoire *DIC9150*.
2. Démarrez `jupyter-lab` sur la ligne de commande et ouvrez *mspEx.ipynb*.
3. Nous allons lire, grossièrement, le code ensemble pour comprendre ce qui se passe.
4. Observez que les contraintes exprimées sont celles présentées dans les acétates.
5. Faites maintenant exécuter chacune des cellules de code avec Ctrl-Entrée.
6. Vérifiez que l'instruction `print(s.model())` affiche bien une solution pour notre problème.
7. Ajoutez une nouvelle cellule avec le + en haut et vérifiez que `s.check(D != 1)` vérifie s'il y a une solution aux contraintes auxquelles est ajoutée la contrainte supplémentaire `D != 1`. S'il y a une solution `sat` s'affiche, sinon `unsat` s'affiche. Quel résultat obtenez vous? Si vous obtenez une solution, vous pouvez la faire afficher en réexécutant `s.model()`. S'il y a une solution, vérifiez donc si elle satisfait la contrainte additionnelle.
8. Utilisez de nouveau l'instruction `s.check()`, mais cette fois-ci pour indiquer que *B* et *D* ne sont pas voisins. Vérifiez si vous obtenez une solution. **Indice** : plusieurs contraintes peuvent être mises dans `s.check()` en les séparant par des virgules.

1. ou sinon `pip install jupyterlab`

