

PROJET SDD L2 2015 : Solveur de Rubik's Cube

Franck Lepoivre

28 octobre 2011

1 Sujet

Version RFC : pour questions, commentaires et demandes de précisions.

Concevoir un solveur de Rubik's Cube¹.

Vous avez pleine liberté dans l'élaboration de vos algorithmes. En revanche, les SDD de base (états et permutations) vous sont imposées.

Votre travail sera apprécié principalement en termes de performance logique (qualité de la solution = minimalité du chemin de résolution) et de performance physique (temps de calcul de la solution).

2 Modélisation des états et des transitions

2.1 Structure algébrique

Un Rubik's Cube est composé de 6 faces chacune composée de 9 facettes. Ce cube peut être transformé à l'aide de 6 opérations de base² que l'on peut enchaîner. L'ensemble de ces 6 opérations est le générateur d'un *groupe d'action* qui est l'ensemble des transformations licites du Rubik's Cube à partir d'un état initial. L'ensemble des états du Rubik's Cube est donc le cardinal de ce groupe d'action.

2.2 Convention, notations

Commençons par associer un index à chacune des 6 faces du cube en état résolu. Les 6 faces sont respectivement colorées en blanc, bleu, rouge, orange, vert, jaune. On associe respectivement à ces 6 couleurs les identifiants 1 à 6.

2.2.1 Repère

Le cube est tenu face à soi, face blanche au premier plan, face jaune à l'arrière plan, face rouge au dessus, face orange en dessous, face bleue sur le côté gauche, face verte sur le côté droit. En dérivent trois axes de rotation perpendiculaires à chacun des couples de faces opposées, traversant leur facette centrale :

- l'axe x horizontal, orienté de gauche à droite, qui traverse les faces bleue et verte,
- l'axe z vertical, orienté de bas en haut, qui traverse les faces orange et rouge,

1. Ce sujet est l'adaptation d'un sujet de TP L3 donné l'année dernière. Les L3 n'avaient que 5 jours, et il leur était imposé d'utiliser l'algorithme A*. Vous avez environ 45 jours, et aucun algorithme particulier ne vous est imposé : soyez inventif, ingénieux, et surtout.. ne vous y prenez pas au dernier moment

2. 6 rotations d'un quart, dans le sens trigonométrique pour chacune des 6 faces

- l'axe y de profondeur, orienté de l'arrière à l'avant, qui traverse les faces jaune et blanche,

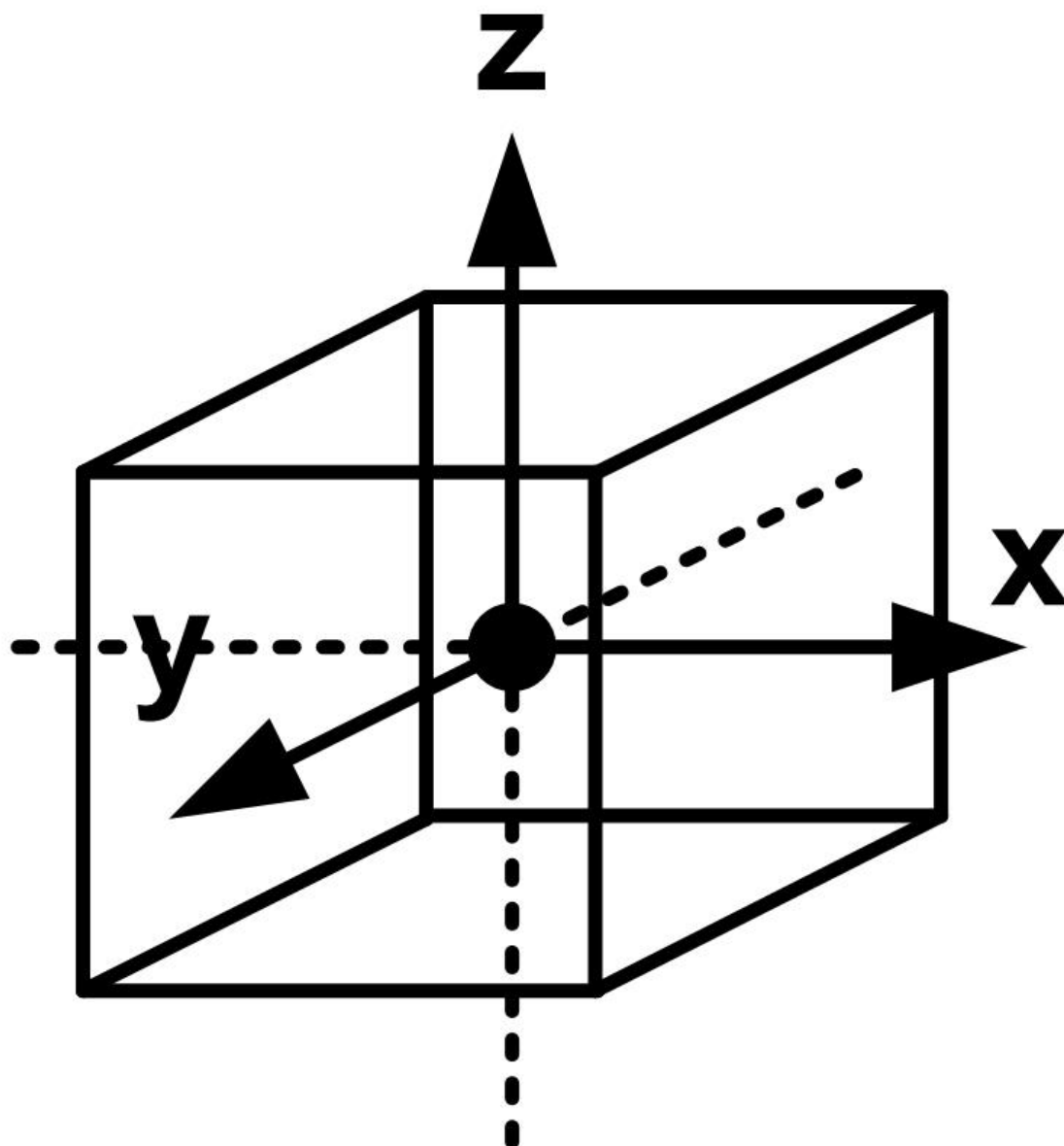


FIGURE 1 – Repère

2.2.2 Opérations

Définissons à présent les 6 opérations élémentaires :

- Rotation d'un quart ω_z de la rangée supérieure : c'est la rotation à raison d'un angle droit, dans le sens trigonométrique selon l'axe z , de la face rouge tenue face à soi.
- Rotation $\omega_{\bar{z}}$ d'un quart de la rangée inférieure : c'est la rotation à raison d'un angle droit, dans le sens trigonométrique selon l'axe z , de la rangée orange tenue face à soi.
- On définit de la même manière les opérations $\omega_x, \omega_{\bar{x}}, \omega_y, \omega_{\bar{y}}$.

2.3 Structure de donnée pour représenter un état

La théorie des groupes d'actions nous invite à représenter un état comme un n -uplet, et une action comme l'application d'une permutation à ce n -uplet.

En l'occurrence, un état est un 54-uplet avec 9 répétitions de chacune des 6 couleurs que l'on peut représenter à l'aide d'un simple tableau de taille 54.

Chaque action est une permutation du groupe symétrique d'ordre 54.

2.3.1 Indexation

Poursuivons notre travail de convention en définissant le 54-uplet comme la séquence de 6 blocs des 9 facettes qui occupent respectivement les faces 1 à 6. Pour chacun de ces blocs, considérons les 9 facettes dans l'ordre de parcours en largeur, de haut en bas, de gauche à droite.

Pour chaque face considérée, précisons que les deux axes bas vers haut et gauche vers droite sont les deux axes pris parmi x, y et z , parallèles à la face, formant nécessairement un angle d'un quart, tels que, suivant le sens trigonométrique, le premier joue le rôle d'axe gauche vers droite, le second joue celui d'axe bas vers haut.

En utilisant pour plan de numérotation des places un patron d'assemblage du Rubik's Cube en état résolu, les conventions précédentes donnent ceci :

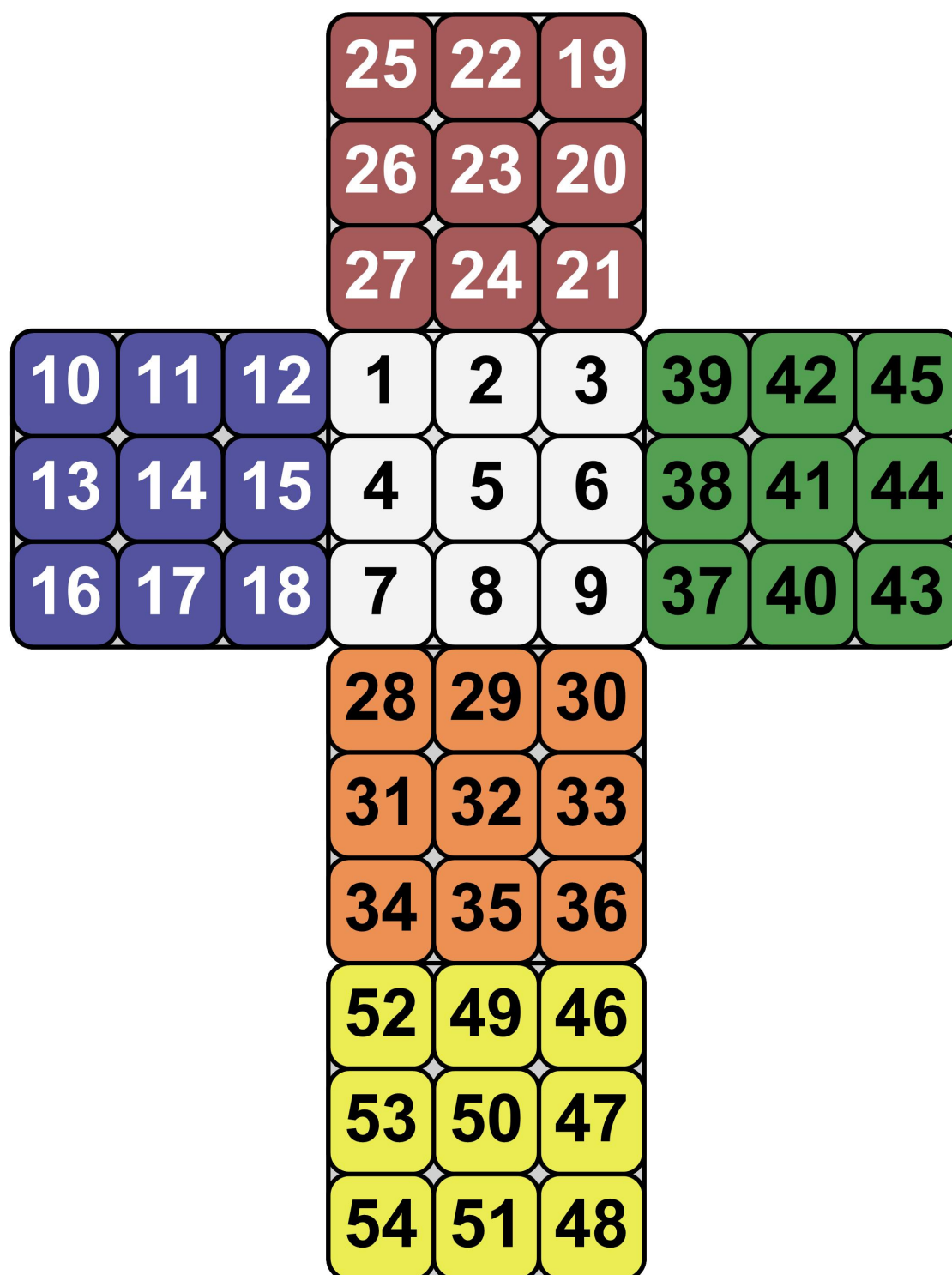


FIGURE 2 – Indexation des facettes

Ces conventions nous permettent de représenter tous les états du Rubik's Cube ainsi que les opérations qui permettent de transiter entre ces états.

Considérons par exemple, la rotation ω_x appliquée au Cube depuis son état résolu :

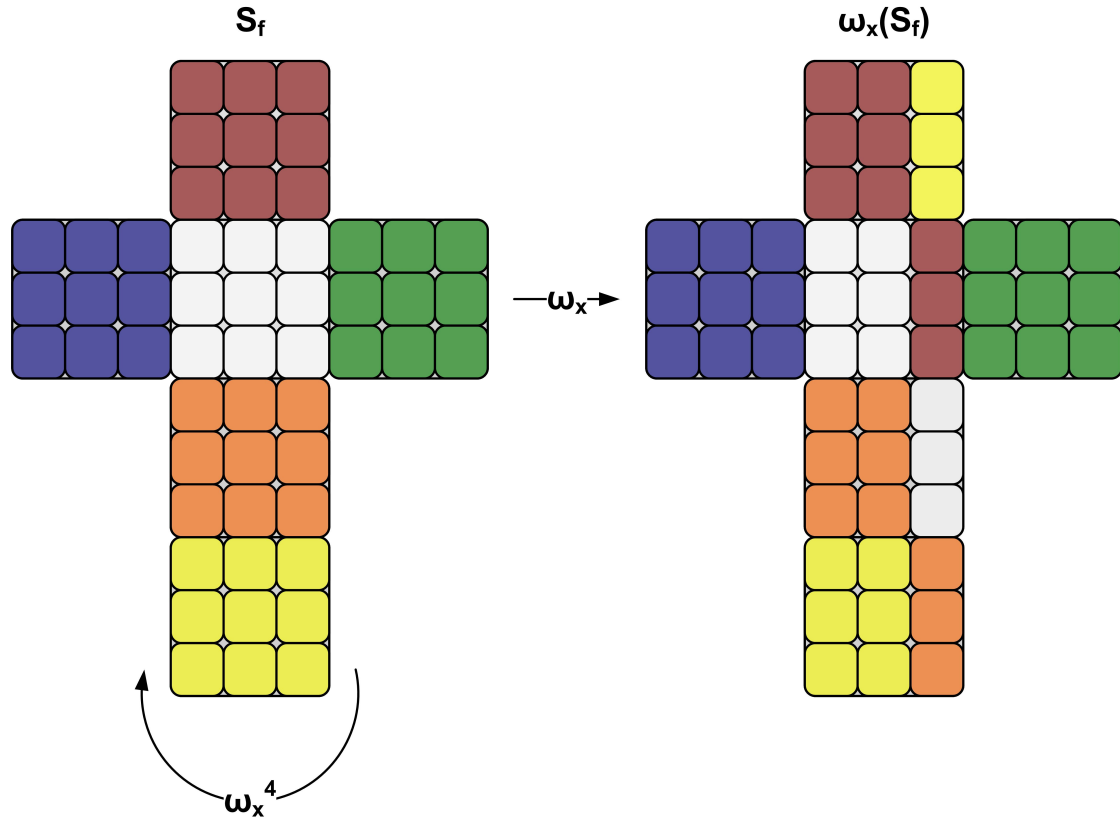


FIGURE 3 – Rotation

L'état du Cube résolu se représente par :

111111112222222233333333444444445555555566666666

L'état engendré par la rotation ω_x se représente par :

1131131132222222266633333344144144155555555444666666

La rotation ω_x est une permutation (des composantes du vecteur d'état) que l'on peut exprimer sous forme de décomposition en produit de cycles³ comme :

$$(3 \ 30 \ 46 \ 19) (6 \ 33 \ 47 \ 20) (9 \ 36 \ 48 \ 21) (39 \ 37 \ 43 \ 45) (38 \ 40 \ 44 \ 42)$$

C'est la permutation qui amène en position 3 la facette de position 19, en position 19 celle de position 46, en position 46 celle de position 30, en position 30 celle de position 3, ...

3. Il sera utile de vous documenter sur cette notion

Notons que cette permutation comporte 34 points fixes.

Les SDD suivantes vous sont imposées :

- tableau de taille 54 pour représenter un état du Rubik's Cube.
- LSC de LSC circulaires pour représenter une permutation (cf. produit de cycles).

3 Cas à résoudre et trace d'exécution

Fournir une trace d'exécution explicite des étapes de votre algorithme notamment des ajouts de nouveaux états dans l'ensemble des ouverts avec évaluation de leurs coût et heuristique, le passage d'un état des ouverts dans les fermés.

Votre programme démarre en chargeant l'état initial du Rubik's à partir d'un fichier nommé `input.txt` et produit un fichier `output.txt`.

Ces fichiers contiennent chacun une séquence d'opérations élémentaires, le premier, celle qui a permis de le « mélanger » depuis l'état résolu, le second, celle qui a permis de le résoudre.

Il est impératif de respecter la table de codage des opérations élémentaires fournie en annexe. Contenu de test suggéré pour `input.txt` :

JHGLDZGLYZ@DPU@IHKGRJT@AWLCVXZSIASMYKCU@HGAXLMMWDDPNPTEKXR@

4 IHM et technologies tierces parties

L'appréciation de l'IHM n'est qu'un constituant accessoire de la note. Évitez d'y investir trop de temps, au détriment de la recherche de la meilleure performance algorithmique.

Vous pouvez utiliser les technologies de votre choix.

Les technologies tierces partie employées devront être dûment référencées dans votre livrable, et notamment rappelées dans les annexes de votre rapport.

Il est interdit d'utiliser des bibliothèques existantes pour vos SDD et algorithmes clés (en d'autres termes, pas de librairie, sauf librairie graphique).

5 Rapport

- Un sommaire détaillé avec pages numérotées.
- L'analyse fonctionnelle générale (environ cinq pages) : Elle explique quelles sont les principales données traitées et quelles sont les principales fonctionnalités du programme, comment on peut les regrouper, comment elles sont organisées dans le programme. Cette analyse aboutit à la présentation des modules fonctionnels qui sont décrits en détail dans l'analyse détaillée. L'analyse générale doit préciser le ou les choix importants que vous avez faits pour la conception de votre projet.
- L'analyse fonctionnelle détaillée (environ dix pages) : Cette analyse précisera pour chaque module son utilité, ses entrées et ses sorties, les données qu'il modifie. Il faut décrire les données utilisées et les algorithmes mis au point (logique de manipulation) avec rigueur, concision et clarté, accompagnée de schémas. C'est un travail de synthèse, de pédagogie et de communication : les extraits de code sont proscrits de vos rapports.
- Annexes : au moins les technologies tierces employées et conseillé, bibliographie et webographie.

6 Éléments logistiques

Binômes par groupe de TD. Les délégués de groupe sont chargés de produire une liste des équipes et des ordres de passage sur les plages horaires réservées pour les soutenances. En cas de dysfonctionnement problématique au sein d'une équipe, n'attendez pas le dernier moment pour l'évoquer : on peut toujours trouver une solution.

7 Livraison

Pour chaque équipe, remise des travaux (une seule par équipe) le DATE.

Envoi par email à `franck.lepoivre@gmail.com`, en précisant `PROJET SDD L2 GP - NOM 1 - NOM 2.RAR` dans l'objet.

Forme et contenu du livrable : une archive RAR nommée `PROJET SDD L2 GP - NOM 1 - NOM 2.RAR` contenant votre code source, votre documentation au format PDF, pas d'exécutable, ce qu'il faut pour compiler.

7.1 Date de livraison

Le DATE à HH :MM dernier délai. Au delà, une pénalité de 2 points par jour de retard.

8 Annexe - table des opérations élémentaires

Notation	Code	Définition	Commentaire
ω_x	A		Rotation avant d'un quart de la rangée supérieure de l'axe des x
ω_y	B		...
ω_z	C		...
$\omega_{\overline{x}}$	D		
$\omega_{\overline{y}}$	E		
$\omega_{\overline{z}}$	F		
$\overline{\omega_x}$	G	ω_x^3	
$\overline{\omega_y}$	H	ω_y^3	
$\overline{\omega_z}$	I	ω_z^3	
$\overline{\omega_{\overline{x}}}$	J	$\omega_{\overline{x}}^3$	
$\overline{\omega_{\overline{y}}}$	K	$\omega_{\overline{y}}^3$	
$\overline{\omega_{\overline{z}}}$	L	$\omega_{\overline{z}}^3$	
ω_{x^2}	M	ω_x^2	
ω_{y^2}	N	ω_y^2	
ω_{z^2}	O	ω_z^2	
$\omega_{\overline{x}^2}$	P	$\omega_{\overline{x}}^2$	
$\omega_{\overline{y}^2}$	Q	$\omega_{\overline{y}}^2$	
$\omega_{\overline{z}^2}$	R	$\omega_{\overline{z}}^2$	
ω_{x^*}	S	$\overline{\omega_x} \omega_{\overline{x}}$	
ω_{y^*}	T	$\overline{\omega_y} \omega_{\overline{y}}$	
ω_{z^*}	U	$\overline{\omega_z} \omega_{\overline{z}}$	
$\overline{\omega_{x^*}}$	V	$\omega_x \overline{\omega_{\overline{x}}}$	
$\overline{\omega_{y^*}}$	W	$\omega_y \overline{\omega_{\overline{y}}}$	
$\overline{\omega_{z^*}}$	X	$\omega_z \overline{\omega_{\overline{z}}}$	
ω_{x^*2}	Y	$\omega_{x^*}^2$	
ω_{y^*2}	Z	$\omega_{y^*}^2$	
ω_{z^*2}	$@$	$\omega_{z^*}^2$	