

Présentation de Sudokusc

On souhaite concevoir Sudokusc, un programme de résolution des grilles de Sudoku. Pour ce faire, il nous faut trouver un algorithme de résolution, choisir un langage pour l'implémenter, rédiger le code et tester le programme.

1 L'algorithme

L'algorithme doit être :

- Simple à expliquer à un ordinateur (répétition d'étapes simples un grand nombre de fois) ;
- Rigoureux et précis (pas question de laisser des zones d'ombre) ;
- Exhaustif (résolution de toutes les grilles) ;
- Rapide (autant que possible) ;

J'ai choisi un algorithme *récuratif* : on explore l'arbre des solutions possibles pour la grille en complétant chaque case au hasard parmi les chiffres qui peuvent légalement s'y trouver, et on recommence à partir des grilles ainsi obtenues. Quand on est coincé, on fait marche arrière. Quand la grille est pleine, on a gagné !

2 Le langage

On doit rédiger cet algorithme dans un langage de programmation (on obtient le *code*) et la machine le transformera en instructions qu'elle peut comprendre.

Un langage est dit *compilé* s'il traduit le code en instructions une seule fois, et *interprété* s'il le fait à chaque fois. Les langages compilés sont généralement plus rapides (pas besoin de recommencer la même chose à chaque fois) mais il faut compiler pour chaque machine.

Un langage est dit *haut niveau* s'il est proche du langage humain, et *bas niveau* s'il est proche de celui de l'ordinateur. Les langages haut niveau ont une syntaxe plus naturelle mais moins rigoureuse, plus concise mais plus désordonnée, ils sont moins rapides mais plus simples, plus indépendants de la machine mais permettent de faire moins de choses.

J'ai choisi le langage C. Il est bas niveau, compilé, très rapide, très rigoureux (mais inclut peu de "garde-fous"), normalisé ANSI et ISO, vieux (1972, normalisé en 89 et a évolué depuis), très basique dans sa syntaxe, la base de quelques « extensions » (C++, etc.), *portable* (indépendant de la machine) si l'on fait attention à ce que l'on fait, et déjà très utilisé.

Les pointeurs sont *primordiaux* en C (et Sudokusc s'en sert beaucoup), ce sont des adresses mémoire pointant vers une autre adresse.

3 Le programme

Le programme obtenu est court (6000 caractères dont environ 4000 de commentaires non intégrés au résultat final), rapide (résout un Sudoku en quelques millisecondes), fonctionnel en dimension N (super-sudoku, etc.), libre (sous GNU GPL), mais pas encore très convivial (terminal).