

**Jeu du Binero exprimé sous forme d'un
problème de satisfaction de contraintes booléennes**

A rendre pour le 25 mai - Date de soutenance envisagée le 28 mai

Pascale Le Gall

1 Avant-propos : consignes générales

- Les devoirs peuvent être faits individuellement ou en binôme, avec une préférence marquée pour la configuration binôme.
- Le rendu d'un devoir de programmation se présente sous la forme d'un répertoire compressé comprenant, à titre indicatif, l'ensemble des fichiers suivants :
 - Un fichier `readme.txt`** qui liste l'ensemble des fichiers du répertoire en indiquant le rôle de chacun des fichiers ou répertoires présents ;
 - Le ou les fichiers de code source** (par défaut, chaque fichier source contient un commentaire en-tête indiquant le rôle du fichier, l'auteur et les dates de création, et de révision successives, et il est attendu que le code soit commenté) ;
 - Le ou les fichiers de données utilisées** ainsi que le ou les fichiers de résultats, pour peu qu'ils présentent un intérêt ;
 - Un mode d'emploi simple** pour exécuter les programmes, sur vos données de test (faire en sorte qu'une commande sans arguments permette d'exécuter vos programmes et de fournir les résultats assortis de quelques commentaires) ;
 - Un document rédigé**, en général sous format pdf comprenant les informations suivantes :
 - L'architecture générale du projet (à savoir l'organisation des principales fonctions, types de données, fichiers) ;
 - Les principaux choix de conception effectués ;
 - Les limitations adoptées dans le projet ;
 - Les réponses aux questions directement incluses dans l'énoncé (tout particulièrement la question relative à la mise sous forme normale conjonctive) ;
 - Des instructions claires d'utilisation du code ;
 - Des illustrations (le jeu de Binero s'y prête).
- L'exploitation de ressources externes (livres, internet, etc) est autorisée sous réserve d'indiquer les sources. En revanche, il est attendu que deux devoirs rendus ne soient pas des copies conformes l'un de l'autre.
- Le sujet est à géométrie variable :
 - Il est attendu un soin tout particulier à la modélisation du problème sous forme de formules booléennes sous forme normale conjonctive.
 - Les dernières sections (à partir de la section 7) peuvent être abordées de façon succincte, ou même omises.
 - Selon vos affinités, vous avez toute liberté du choix du langage de programmation et du solveur de contraintes. Des exemples de solveurs sont MiniSat, Sat4j, cvc4, Z3, choco, Zchaff, Yices ...¹
Ces solveurs sont soit des bibliothèques fournies par certains langages (python-constraint pour Python 2.x ou bien le solveur Z3 pour Python 3.x par exemple) ou indépendants avec des formats de données spécifiques. En particulier, les solveurs peuvent manipuler d'autres formules que les

1. <http://www.satcompetition.org/>

simples formules booléennes en fnc : on parle alors de solveurs modulo théories (SMT). Néanmoins ils incluent a minima les formules booléennes. Dans le cadre de ce projet, même si vous faites appel à un solveur SMT, il est expressément demandé de ne manipuler que des formules booléennes.

- La priorité est d'abord à la qualité de la modélisation avant le passage à l'échelle : vous êtes invités à valider votre approche (modélisation et solution) sur des exemples de taille raisonnable (en particulier sur l'exemple donné dans le sujet). Une fois cette première phase de validation effectuée, vous évalueriez expérimentalement les performances (passage à l'échelle) sur des données de taille plus conséquente.
- N'hésitez pas à poser des questions (par email ou de vive voix) si l'énoncé ne vous paraît pas clair ou tout simplement si vous avez besoin d'éclaircissements.

2 Format DIMACS

Le format DIMACS est un standard pour la représentation des formules booléennes en forme normale conjonctive. Un fichier en format DIMACS se présente sous la forme suivante :

- des lignes de commentaires commençant par la lettre 'c', et donnant les principales informations relatives au fichier (auteur, date, problème modélisé, etc) ;
- une ligne de la forme 'p cnf <num vars> <num clauses>', commençant par la lettre 'p' et indiquant qu'il s'agit d'une forme normale conjonctive ('cnf'), portant sur un certain nombre de variables propositionnelles ('<num vars>') et comprenant un nombre de clauses ('<num clauses>').

Par exemple :

```
p cnf 5 3
```

indique une formule en forme normale conjonctive avec 5 variables et 3 clauses.

Les variables sont numérotées à partir de 1 ;

- et ensuite, autant de lignes qu'il y a de clauses. Chaque ligne contient des entiers séparés par des espaces terminant par 0. Un entier positif i indique que la i -ème variable apparaît avec une polarité positive (littéral positif), tandis qu'un entier négatif $-i$ indique que la i -ème variable apparaît avec une polarité négative (littéral négatif).

Par exemple, la formule

$$(x_1 \vee \neg x_5 \vee x_4) \wedge (\neg x_1 \vee x_5 \vee x_3 \vee x_4) \wedge (\neg x_3 \vee \neg x_4)$$

est encodée dans un fichier en format DIMACS de la façon suivante :

```
c
c un petit exemple
c
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

Les formats de retour sur la console sont de la forme suivante :

```
s UNSATISFIABLE
```

si la formule est insatisfiable, et de la forme suivante

```
s SATISFIABLE
v 1 2 -3 4 5 0
```

si la formule est satisfiable. Dans ce dernier cas, la seconde ligne contient une valuation qui permet d'évaluer à vrai la formule décrite par le fichier DIMACS. Dans l'exemple ci-dessus, la valuation v associant la valeur 1 aux variables x_1 , x_2 , x_4 et x_5 et la valeur 0 à la variable x_3 satisfait la formule décrite par le fichier DIMACS ci-dessus.

3 Modélisation du jeu Binero

3.1 Jeu Binero (ou Takuzu)

Le Binero est un jeu de logique auquel on joue de façon solitaire (comme on joue au Sudoku). Au départ, une grille de Binero se présente sous la forme d'une grille incomplète avec quelques cases remplies de 0 ou de 1 (cf Figure 1 pour un exemple de grille de Binero de taille 8×8).

| | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|
| | | | | | | 1 | |
| | 1 | | | | | | |
| | | 0 | 0 | | 0 | | 0 |
| 1 | | | | | | | |
| | | | 0 | | 0 | | |
| | 0 | | | 1 | | | |
| | | 1 | | | | | 0 |
| | 0 | | 0 | | | | 0 |

FIGURE 1 – Un exemple de grille de binero

Il s'agit de compléter la grille de jeu Binero avec des 0 et des 1 dans toutes les cases en suivant les règles suivantes :

- sur chaque colonne ou ligne de la grille, il ne peut y avoir plus de deux 0 ou deux 1 consécutifs ;
- il y a le même nombre de 0 et de 1 sur chaque ligne et chaque colonne² ;
- il n'y a pas deux lignes (ou deux colonnes) remplies identiquement.

Beaucoup de jeux logiques (comme les Sudoku) peuvent se modéliser sous forme d'une formule en forme normale conjonctive. Dans le cadre de ce projet, il s'agit de modéliser le jeu de Binero (ou Tazuku)³.

Une solution de la grille donnée en exemple en Figure 1 est donnée en Figure 2 : les cases initialement préremplies sont remplies avec des chiffres en noir, tandis que les cases remplies par le joueur sont remplies avec des chiffres grisés. En pratique, de même que lorsqu'on joue au Sodoku, on peut être amené à mettre

2. La taille d'une grille de Binero, que ce soit en longueur ou en hauteur, est donc paire par construction.

3. cf par exemple la page <http://www.sudoku-land.com/binero/binero.php> pour jouer en ligne

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

FIGURE 2 – Solution de la grille donnée en figure ??

un chiffre, disons 0, sur une case c , puis se rendre compte plus tard que ce choix mène à une impasse, et donc revenir sur l’option initiale et mettre alors le chiffre 1 sur la case c (étape dite de backtracking ou retour arrière), afin de réussir à remplir la grille en respectant les règles.

3.2 Format des fichiers du jeu de Binero

Afin de convenir d’un format d’entrée commun, appelé format BINERO, une grille de Binero sera donnée sous la forme d’un fichier texte sous le format suivant :

- des lignes de commentaires commençant par la lettre ‘c’, et donnant les principales informations relatives au fichier (auteur, date, etc) ;
- une ligne de la forme ‘binero <num1> <num2>’, comportant le mot ‘binero’ et deux entiers strictement positifs pairs indiquant respectivement la longueur (<num1>) et la hauteur (<num2>) de la grille. Ainsi, les grilles de Binero peuvent être rectangulaires ou carrés. Dans ce dernier cas, la taille du carré est répétée deux fois (<num1> et <num2> dénotant tous les deux la taille du carré).

On aura ainsi la ligne :

binero 8 8

pour la grille de Binero donnée en exemple dans la Figure 1 ;

- et ensuite, autant de lignes qu’il y a de lignes dans la grille de Binero. Chaque ligne contient une liste de caractères choisis dans {1,0,.}, séparés par des espaces. Les chiffres 0 et 1 indiquent les premières cases remplies de la grille du problème Binero (soit avec un 0, soit avec un 1), tandis que le caractère ‘.’ dénote une place vide que le joueur doit remplir en respectant les règles du jeu, et les contraintes induites par les premiers chiffres 0 et 1 déjà placés.

Par exemple, pour la grille donnée en exemple, cela donne :

```
c
c un petit binero
```

```

c
binero 8 8
. . . . . 1 .
. 1 . . . . .
. . 0 0 . 0 . 0
1 . . . . . .
. . . 0 . 0 . .
. 0 . . 1 . . .
. . 1 . . . . 0
. 0 . 0 . . . 0

```

3.3 Format des fichiers d'un jeu de Binero résolu

Afin de sauvegarder les jeux résolus, les solutions seront données selon le même format (BINERO) :

- des lignes de commentaires commençant par la lettre 'c', et donnant des informations jugées pertinentes ;
- une ligne commençant par le mot 'binero', suivi de deux entiers strictement positifs pairs pour la longueur (<num1>) et la hauteur (<num2>) de la grille ;
- Par ligne de la grille de binero, une liste de caractères choisis dans {1,0}, séparés par des espaces.

Notez que par rapport à une grille de problème Binero, le caractère "." n'est plus autorisé.

Ainsi, la solution donnée en Figure 2 peut être sauvegardée dans le fichier suivant :

```

c
c un petit binero
c
c
binero 8 8
0 0 1 1 0 0 1 1
0 1 0 1 0 1 0 1
1 1 0 0 1 0 1 0
1 0 1 1 0 1 0 0
0 1 0 0 1 0 1 1
1 0 0 1 1 0 0 1
0 1 1 0 0 1 1 0
1 0 1 0 1 1 0 0

```

4 Modélisation d'un jeu de Binero comme une formule fnc

Il s'agit de modéliser un jeu de Binero sous la forme d'une formule en forme normale conjonctive (fnc). Cette modélisation intégrera a minima une variable par case (i, j) de la grille.

L'enjeu est de traduire les règles du jeu de Binero en contraintes booléennes. Il est demandé d'expliquer et commenter la modélisation que vous proposerez.

5 Implémentation de la modélisation des jeux de Binero

Il s'agit d'implémenter la modélisation proposée dans la section précédente, en prenant en entrée un jeu de Binero donné sous forme d'un fichier sous le format BINERO et renvoyant un fichier sous le format DIMACS.

Remarque : aucun langage de programmation n'est imposé. Quel que soit le langage de programmation choisi, le code doit être aussi lisible que possible (en particulier, être annoté de commentaires judicieux).

6 Résolution des jeux de Binero à l'aide de solveurs

En faisant appel au solveur de votre choix (éventuellement votre propre solveur), il s'agit de résoudre un jeu de Binero donné sous forme d'un fichier DIMACS. Il est demandé de sauvegarder la solution sous format BINERO. Néanmoins, à des fins de visualisation (et surtout de débogage), vous pouvez être amenés à afficher la solution sous un format propre (comparable à celui de la Figure 2).

Il est demandé de valider votre approche avec différents exemples de votre choix, que vous pouvez bien sûr choisir parmi ceux proposés sur Internet ⁴.

Remarque : par courtoisie, vous pouvez mettre à disposition des autres groupes des grilles de binero que vous aurez encodé en format BINERO ⁵.

7 Discussion autour de la performance

Une fois que vous aurez validé votre modélisation, vous êtes invités à mener une étude expérimentale sur les performances de votre implémentation en lien avec les paramètres qui jouent sur la performance.

Voici quelques éléments pour alimenter votre étude :

- Comparaison selon la taille des grilles ;
- Comparaison entre différents solveurs ;
- Comparaison entre jeux faciles (beaucoup de cases forcées) et jeux difficiles ;
- ...

A l'issue de cette discussion, il est attendu que vous fournissiez votre jeu de Binero champion (sous format BINERO), c'est-à-dire que vous avez réussi à résoudre et qui sera proposé aux autres groupes comme challenge à battre. Pour affiner la comparaison, il est demandé de fournir en même temps que le jeu champion le temps de calcul nécessaire pour le résoudre sur votre machine ⁶.

8 Implémentation d'une stratégie directe de résolution

Cette question est optionnelle. Il s'agit de :

- proposer une stratégie de résolution directe des jeux de Binero dans le langage de programmation de votre choix en vous inspirant à la fois de l'algorithme DPLL (pour l'aspect combinaison de stratégies élémentaires et backtracking en cas d'impasse) et des stratégies de résolution mises en œuvre par des joueurs *humains*.
- comparer les performances de la stratégie directe avec l'utilisation de solveur de contraintes.

9 Construction de grilles de Binero

Cette question est optionnelle. Il s'agit de proposer la construction de grilles de jeux de Binero. L'enjeu est de construire des grilles, en minimisant le ratio de cases déjà remplies (par des 0 ou des 1) et avec l'objectif qu'il n'y ait qu'une seule solution possible.

4. Le cas échéant, veuillez indiquer la source.

5. Les transmettre par email à pascale.legall@centralesupelec.fr afin qu'elles soient mises sur claroline.

6. Indiquez les caractéristiques clés de votre machine pour faciliter l'analyse des performances