Aabu

Combien y a-t-il de positions au morpion ?

16 septembre 2020

Table des matières

Il y a quelques jours, je me suis dit, et si je m'intéressais aux mathématiques du morpion? Si vous ne savez pas ce qu'est le morpion, il s'agit d'un jeu simple qui se joue sur une grille de 3 par 3 et où les joueurs posent leurs pions alternativement; le jeu se termine par la victoire du premier joueur qui réussit à former un alignement de trois pions (sur une ligne, colonne ou diagonale) ou lorsque la grille est pleine.

Les questions mathématiques sont nombreuses, mais j'ai choisi pour ce billet de m'intéresser aux nombres de *positions* différentes au jeu de morpion. Une position est une configuration de la grille, indépendamment de la séquence de coups qui a permis d'y parvenir. Dans la suite de l'article, je numérote les cases de la grille comme suit:

```
+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+
```

Cette numérotation interdit de considérer comme équivalentes les rotations et les symétries de la grille, qui permettent de réduire le nombre de position véritablement différentes d'un point de vue stratégique.

Plutôt que de présenter un raisonnement bien ficelé *a posteriori*, je vous propose de suivre le raisonnement qui m'a amené à la réponse, et comment se convaincre que cette réponse est effectivement la bonne.

1. Trouver des limites au nombre de positions

Avant de commencer à chercher le nombre exact, je me suis demandé si je ne pouvais pas encadrer le nombre de position par des bornes, afin de se donner une idée de l'ordre de grandeur du nombre de position envisageable.

1.1. Une borne minimale

On sait qu'il y a au moins une position: la position vide. On peut augmenter cette valeur en considérant qu'il y a 9 manières de jouer le premier coup. On a alors 10 positions (1 + 9). Puis pour chacune des 9 positions correspondant au premier coup, on a huit façons de jouer le deuxième coup, donc il y a 72 positions à deux pions (9 x 8). On arrive donc à 82 positions (10 + 72).

On pourrait continuer comme ça encore un moment, puisque ces estimations sont exactes, mais ce n'est pas nécessaire: on voit qu'on a au moins plusieurs dizaines (et probablement plusieurs centaines) de positions. Il ne nous sera pas possible de les énumérer une par une, c'est certain.

1.2. Quelques majorants

On a vu qu'il y avait 9 possibilités pour jouer le premier coup, puis 8 pour le deuxième. Si on continue, on se dit qu'il y en 7 pour le troisième, puis 6 pour le quatrième, etc. On peut continuer comme ça, ce qui revient à calculer la factorielle de 9:

$$9! = 362\,880$$

Le calcul donne bien le nombre de positions exact au départ, puis il devient un majorant, car à partir d'un moment, on peut se rendre compte qu'on a des répétitions de positions: si on joue 1 puis 2 puis 3, ça revient à faire 3 puis 2 puis 1. Il y a aussi des parties terminées qui sont comptées comme si elles continuaient. On compte donc *beaucoup* de positions en double.

On peut aussi voir qu'il y a 9 cases qui peuvent être soit un pion du premier joueur, soit du deuxième joueur, soit vide, ce qui fait $3^9 = 19\,683$ possibilités. On compte plein de choses impossibles, mais c'est déjà mieux.

1.3. Un majorant un peu plus serré

Dans un premier temps, je n'ai pas envie de me fatiguer à gérer les fins de parties. Par contre, j'ai très envie d'éliminer les positions comptées en doubles.

Pour ce faire, j'ai eu envie de faire un raisonnement indépendant de l'ordre. Par exemple, une position à trois pions (2 pour un joueur et 1 pour l'autre), ça revient à compter combien de manières il y a de mettre 2 pions sur les 9 cases de la grille, puis combien il y a de façons de mettre le pion sur les 6 cases restantes. Le calcul de ça est facile pour moi, parce que je connais le concept de *combinaisons*, que j'ai appris au lycée.

On peut dénombrer ces combinaisons en séparant les parties à 0 coup, 1 coup, etc. La formule est en fait toujours la même: on compte les manières de mettre les pions du premier joueur sur les 9 cases, et on multiplie par le nombre de manière de mettre les pions du deuxième joueur sur les cases restantes. Le tableau ci-dessous montre le résultat pour cette technique de majoration.

Coups	Jetons X	Jetons O	Formule du majorant	Valeur du majorant
0	0	0	$\binom{9}{0} \times \binom{9}{0}$	1
1	1	0	$\binom{9}{1} \times \binom{8}{0}$	9
2	1	1	$\binom{9}{1} \times \binom{8}{1}$	72
3	2	1	$\binom{9}{2} \times \binom{7}{1}$	252
4	2	2	$\binom{9}{2} \times \binom{7}{2}$	756
5	3	2	$\binom{9}{3} \times \binom{6}{2}$	1260
6	3	3	$\binom{9}{3} \times \binom{6}{3}$	1680
7	4	3	$\binom{9}{4} \times \binom{5}{3}$	1260
8	4	4	$\binom{9}{4} \times \binom{5}{4}$	630
9	5	4	$\binom{9}{5} \times \binom{4}{4}$	126
Total	-	-	-	6046

On se retrouve donc ici avec un majorant qui vaut 6046. C'est beaucoup mieux que ce qu'on avait avant.

2. Cheminer vers la réponse exacte

2.1. Nombre de positions en début de partie

Notre meilleur majorant est vraiment pas mal. En effet, tant qu'il n'est pas possible de gagner, il donne le nombre exact de positions! Il est assez facile de s'en convaincre, puisque tant qu'aucun joueur ne peut gagner, il n'est pas possible d'obtenir des situations invalides du type «un joueur joue alors que l'autre a déjà gagné». L'idée de prendre un majorant qui ne se préoccupe pas des fins de partie était donc une bonne idée pour gérer le nombre exact de position en début de partie.

Avec ça, on se rend compte que le majorant est exact pour tous les nombres de coups de 0 à 4. On peut aussi se rendre compte qu'avec 5 pièces, on peut aussi jouer comme si de rien était. C'est au coup d'après que ça se compliquera, puisque le premier joueur peut avoir gagné.

On sait donc ça pour le moment:

Coups	Nombre exact de positions
0	1

1	9
2	72
3	252
4	756
5	1260

2.2. Nombre de positions à 6 pions

Pour 5 coups, le premier joueur peut gagner. Il faut donc réfléchir à comment éviter de prolonger des parties qui ont déjà été gagnées lorsqu'on traite le cas à 6 coups. Pour ça, il faut calculer le nombre de manière de mettre les trois pions du joueur sans gagner, puis multiplier ça par le nombre de manière de placer ensuite les trois pions du deuxième joueur.

Plutôt que de réfléchir comment compter directement les façons de placer les pions sans gagner, j'ai préféré compter les manières de gagner, et les soustraire au nombre de manière de placer les pions. Pour gagner, il faut placer trois pions alignés sur une des colonnes, lignes ou diagonales. Il y a donc huit positions possibles pour gagner. Ainsi, le nombre de dispositions non-gagnantes pour les trois pions du premier joueur sera $\binom{9}{3} - 8$.

On multiplie par le nombre de manière de placer les trois pions du deuxième joueur et on obtient alors le nombre de positions à 6 pions:

$$\left(\binom{9}{3} - 8\right) \times \binom{6}{3} = 1520$$

2.3. Nombre de positions à 7 pions

Pour celui-là, j'ai été un peu embêté. Au départ, je voulais procéder comme pour les positions à 6 pions, en me disant qu'il suffit d'éviter les positions où le deuxième joueur aurait déjà gagné, mais en faisant attention à ne pas prendre en compte les cas où le premier joueur aurait ses pions alignés aussi. Je me suis embrouillé, et je suis allé nulle part avec cette idée.

Au bout d'un moment, je me suis rendu compte de quelque chose: on peut faire comme si le deuxième joueur avait posé ses 3 pions *sans gagner* et poser les 4 pions du premier joueur comme on veut!

Ce calcul est beaucoup plus simple. On connaît déjà le nombre de façon de poser trois pions sans gagner: $\left(\binom{9}{3} - 8\right)$. Le nombre de manière de poser les quatre pions du premier joueur est alors: $\binom{6}{4}$. Finalement, le nombre de positions à 7 pions est

$$\left(\binom{9}{3} - 8\right) \times \binom{6}{4} = 1140$$

Jusque-là nos calculs semblent justes, et en plus on est bien inférieur à notre majorant, donc ça sent bon la victoire.

2.4. Nombre de positions à 8 pions et à 9 pions

Je suis resté bloqué longtemps pour trouver la solution à 8 pions. En fait, j'étais persuadé que faire comme si les joueurs avaient joué dans le désordre était une astuce qui ne fonctionnait plus ensuite, puisque les deux joueurs étaient en mesure d'avoir gagné (ils ont joué chacun trois pions ou plus). J'ai erré assez longtemps sur des idées vaines.

Au bout d'un moment, j'ai compris la raison pour laquelle ça marchait aux deux étapes d'avant: *le joueur dont ce n'est pas le tour ne peut pas avoir gagné*, peu importe s'il joue en premier ou en deuxième. Cela veut aussi dire que l'idée pour 6 et 7 pions marche pour 8 pions (et 9 pions). Il faut juste savoir compter les positions non-gagnantes.

À huit pions, il faut compter les positions non-gagnantes du premier joueur, qui a 4 pions. On procède pareil en comptant les positions gagnantes d'abord. On a vu qu'il y avait 8 façons de poser 3 pions pour gagner; il reste alors à chaque fois 1 pion à placer sur 6 cases vides. On a donc $6 \times 8 = 48$ façons de gagner.

On déduit de ça le nombre de positions non-gagnantes, comme auparavant, puis on place les pions de l'autre joueur, comme avant. La formule est dans ce cas:

$$\left(\binom{9}{4} - 48\right) \times \binom{5}{4} = 390$$

On raisonne similairement à 9 pions:

$$\left(\binom{9}{4} - 48\right) \times \binom{5}{5} = 78$$

Tous ces nombres sont bien inférieurs au majorant, ce qui est rassurant.

2.5. Tableau final

On arrive au tableau ci-dessous.

Coups	**Nombre exact de positions**
0	1
1	9
2	72
3	252
4	756
5	1260
6	1520
7	1140

8	390
9	78
Total	5478

3. Se convaincre qu'on ne s'est pas trompé

À partir de là, comment savoir si je ne me suis pas trompé?

Pour les positions à 0, 1 et 2 pions, ce n'est pas difficile, parce qu'il n'y a pas d'écueils. Pour celles sans victoires, je suis assez convaincu de ce que j'ai fait, mais qui sait, peut-être que j'aurai pu avoir fait fausse route. Pour les combinaisons avec victoires possibles, ma conviction est encore un peu plus faible...

3.1. Confirmation indépendante

Première technique pour vérifier son résultat: chercher quelqu'un qui l'a déjà fait[1]. J'ai réussi à trouver une page ayant traité la question, avec beaucoup de détails et qui va même encore plus loin. Et ce qui est top, c'est qu'on a tous les deux trouvé la même chose!

3.2. Programme informatique

Vu la taille du problème (bornée par un majorant très bas), il est possible de réaliser un programme informatique qui liste toutes les positions distinctes et les compte. L'avantage de cela, c'est qu'il est possible (moyennant un peu de soin dans la programmation) de faire un comptage très direct; il n'y aura pas d'astuces, puisque il s'agira seulement de lister toutes les parties et de les compter de la manière la plus bête qu'il soit.

Le programme est assez court (et fait un peu à l'arrache), je le mets ci-dessous. Il donne bien la bonne réponse pour le nombre de positions totales.

Contenu masqué n°1

Et voilà! Il y a encore beaucoup à dire sur les mathématiques du morpion, mais... ce sera pour une prochaine fois.

Miniature du billet: illustration d'une partie de morpion par Symode09, domaine public (source).

1. Et pour citer approximativement un de mes profs de maths: «Tous les exercices que nous faisons sont faciles, car quelqu'un les a déjà résolus.»

Contenu masqué

Contenu masqué n°1

```python
P1 = 1
P2 = -1
EMPTY = 0

class Position:
    def __init__(self, pos_vec):
        self.pos_vec = pos_vec

    def allowed_moves(self):
        if self.is_endgame():
            return []
        else:
            return [i for i in range(len(self.pos_vec)) if
                self.pos_vec[i] == EMPTY]

    def current_player(self):
        if len([p for p in self.pos_vec if p == EMPTY]) % 2 == 0:
            return P2
        else:
            return P1

    def next_player(self):
        if self.current_player() == P1:
            return P2
        else:
            return P1

    def play(self, player, move):
        new_pos_vec = [p for p in self.pos_vec]
        new_pos_vec[move] = player
        return Position(new_pos_vec)

    def is_endgame(self):
        p = self.current_player()
        if ((self.pos_vec[0] == p and self.pos_vec[1] == p and
            self.pos_vec[2] == p)
           or (self.pos_vec[3] == p and self.pos_vec[4] == p and
               self.pos_vec[5] == p)
           or (self.pos_vec[6] == p and self.pos_vec[7] == p and
               self.pos_vec[8] == p)
           or (self.pos_vec[0] == p and self.pos_vec[3] == p and
               self.pos_vec[6] == p)
           or (self.pos_vec[1] == p and self.pos_vec[4] == p and
               self.pos_vec[7] == p)
```

```python
           or (self.pos_vec[2] == p and self.pos_vec[5] == p and self.pos_vec[8] == p)
           or (self.pos_vec[0] == p and self.pos_vec[4] == p and self.pos_vec[8] == p)
           or (self.pos_vec[2] == p and self.pos_vec[4] == p and self.pos_vec[6] == p)
          ):
           return True
        elif [p for p in self.pos_vec if p == EMPTY] == []:
           return True
        else:
            return False

    def __hash__(self):
        return hash(tuple(self.pos_vec))

    def __eq__(self, other):
        return self.pos_vec.__eq__(other.pos_vec)

def generate_next_positions(current_position):
    allowed_moves = current_position.allowed_moves()
    player = current_position.next_player()
    next_positions = []
    for move in allowed_moves:
        np = current_position.play(player, move)
        next_positions.append(np)
    return next_positions


def generate_positions():
    empty_board = Position([EMPTY, EMPTY, EMPTY, EMPTY, EMPTY, EMPTY, EMPTY, EMPTY, EMPTY])
    all_positions = {empty_board}
    def gp(current_position):
        nps = generate_next_positions(current_position)
        for p in nps:
            if p not in all_positions:
                all_positions.add(p)
                gp(p)
    gp(empty_board)
    return all_positions

def count_positions():
    all_positions = generate_positions()
    return len(all_positions)


p = count_positions()
print(p)
```

Contenu masqué

Retourner au texte.