Introduction à l'analyse rétrograde

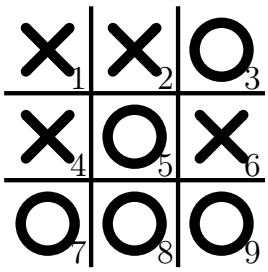
Alain Brobecker - septembre 2023 - http://abrobecker.free.fr/

Avec cet article nous allons nous transformer en explorateurs du temps, tout du moins en pensées et dans le cadre bien restreint des jeux combinatoires. Il faudra se tourner vers le roman de H. G. Wells ou utiliser son imagination pour des aventures plus exotiques.

L'analyse rétrograde propose d'étudier l'histoire qui a permis d'arriver à une situation donnée. Nous en montrons ici deux aspects: les problèmes d'analyse rétrograde, dans le jeu d'Échecs et d'autres jeux, et l'algorithme d'analyse rétrograde (backward induction) qui est utilisé pour automatiser la résolution de situations de jeux combinatoires, voire pour créer des situations problèmes. Les deux aspects sont liés par une définition possible de l'analyse rétrograde: "méthode de résolution d'un problème en partant de la solution finale et en explorant l'arbre de jeu à l'envers".

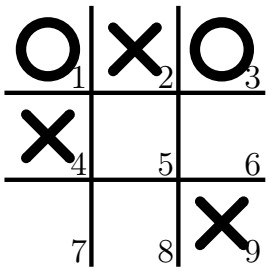
Problèmes d'analyse rétrograde au morpion et à Othello

① **AB**, 2007



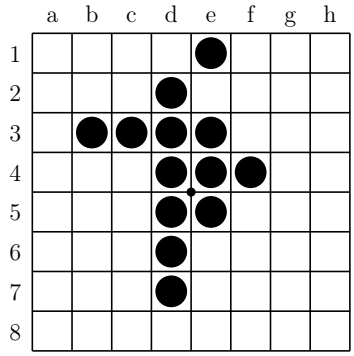
Quel a été le dernier coup joué?

② **AB**, 2007



Personne n'a joué de coup perdant. Comment s'est déroulée la partie?

③ **AB**, 2006



Retrouver la partie.

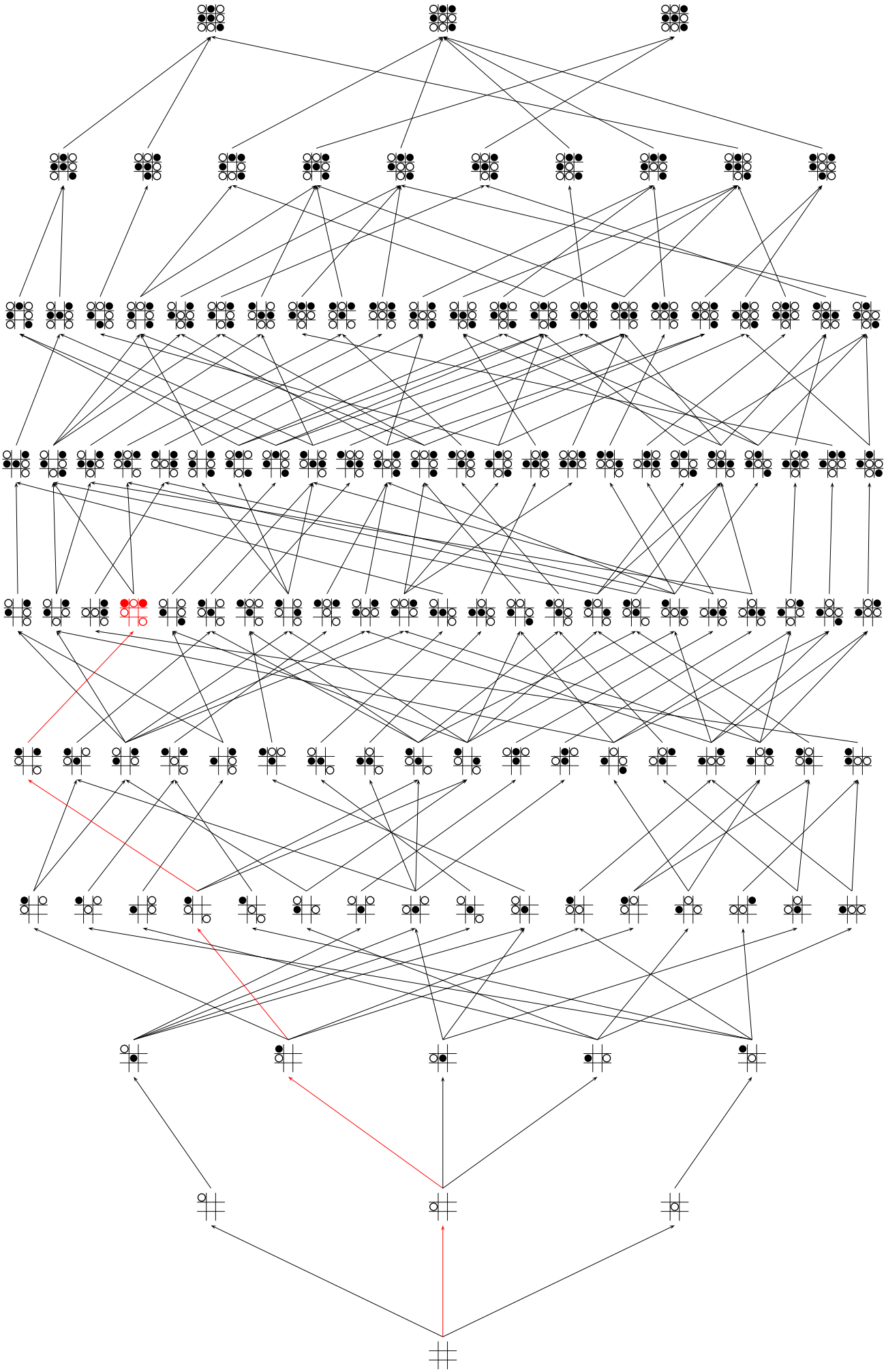
Commençons par résoudre les trois problèmes d'analyse rétrograde qui vous sont proposés ci-dessus. Résoudre un de ces défis nécessite dans un premier temps de connaître les règles du jeu. Intéressons nous donc au morpion (ou tic-tac-toe): le jeu se pratique à 2 joueurs sur une grille de 3 cases par 3 cases, un joueur prend les ronds, l'autre les croix. Ils placent alternativement un de leur symbole sur une case encore vide. Leur but est d'aligner 3 de leurs symboles horizontalement, verticalement ou sur une des diagonales, ce qui met fin immédiatement à la partie.

Dans le ① il est alors aisé de voir que O a joué en dernier puisqu'il a joué 5 symboles contre 4 pour X. De plus O a gagné avec deux alignements, ces deux alignements ont alors dû être réalisés en même temps, O vient donc de jouer à la position 7.

Dans le ② il faut d'abord savoir qu'au Morpion le résultat normal entre deux joueurs expérimentés est un match nul. X a joué en dernier puisqu'il a placé 3 symboles contre 2. Il n'a pas pu jouer en 9 (car en jouant en 5 à la place il gagnait la partie, son adversaire aurait donc joué un coup perdant précédemment) ni en 4 (car 8 à la place gagnait la partie). Il a donc joué son dernier coup en 2. X n'a pas pu jouer son premier coup en 9 car la réponse non perdante de O aurait dû être 5 qui n'a pas été joué. X a donc joué 4, 9 et 2 dans cet ordre. Si O avait joué son premier coup en 3 cela aurait été un coup perdant (par exemple 4, 3?, 1, 7, 5...) donc il a d'abord joué en 1. La partie est donc: 4, 1, 9, 3, 2.

Pour composer le problème ② et d'autres du même acabit, qui sont inspirés d'un problème de Les Marvin (voir tictactoe.pdf), j'ai créé l'arbre contenant les positions pouvant être atteintes quand personne ne joue de coup perdant, puis j'ai cherché les positions ayant une seule prédécesseure (une seule flèche y mène). Le problème ② et la partie associée sont en rouge sur la page suivante.

Les positions du morpion qui peuvent être atteintes lorsque personne ne joue de coup perdant.



L'excellent jeu d'Othello, appelé aussi Reversi, se joue sur un plateau de 8×8 cases avec 64 pions réversibles blancs/noirs. Au début de la partie deux pions noirs sont en d5 et e4, deux pions blancs sont en d4 et e5, et les noirs commencent. À tour de rôle chaque joueur pose un pion de sa couleur de manière à encadrer des pions adverses entre deux de ses pions. Les pions encadrés sont alors capturés par retournement. Si un joueur ne peut pas respecter cette règle il passe son tour. Si aucun joueur ne peut jouer, celui possédant le plus de pions de sa couleur sur le plateau gagne la partie.

La position du ③ est un des gains les plus rapides possibles à Othello, et de manière surprenante il n'y a qu'une seule manière d'aboutir à cette position! Noir a joué en b3, d7, e1, f4 car ces pions n'étant pas encadrés, ils n'ont pas pu changer de couleur. Noir a commencé par 1.d3, on en déduit que blanc a joué c3, d2, d6 et e3, puis après quelques essais on trouve l'intégralité de la partie: 1.d3 c3 2.b3 d2 3.e1 d6 4.d7 e3 5.f4.

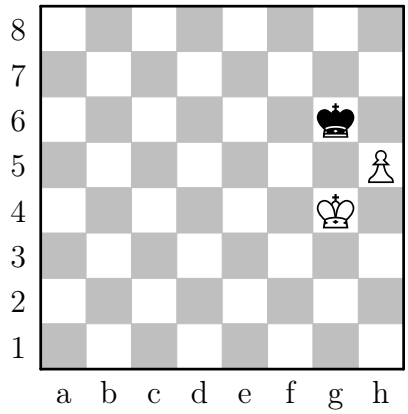
Ce problème et quelques autres (voir othello.pdf), sont inspirés d'un problème d'Erich Friedman trouvé sur une page web de Joseph Kisenwether dédiée à l'analyse rétrograde dans des jeux autres que les Échecs.

L'arbre de jeu d'Othello est hors de portée de la capacité des ordinateurs actuels, pour créer ce problème j'ai programmé un logiciel qui cherche en avant à partir de la position de départ, qui sauve les positions au fur et à mesure qu'elles sont rencontrées et qui compte le nombre de fois ou elles étaient rencontrées à cette profondeur. On peut donc créer un problème d'analyse rétrograde à l'aide d'une recherche en avant, ce qui est agréablement paradoxal.

Problèmes d'analyse rétrograde aux Échecs

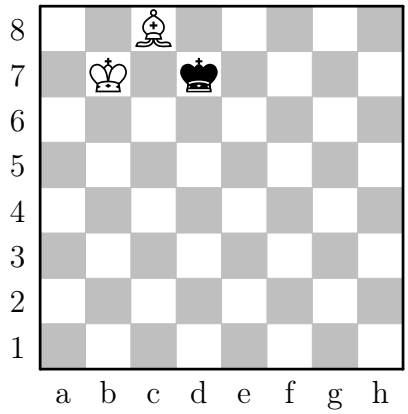
Continuons avec trois problèmes d'analyse rétrograde aux Échecs. Le jeu d'Échecs demande plus d'efforts et je ne vais pas en rappeler les règles ici, mais il recèle une telle richesse dans les parties ou les problèmes que je ne peux que vous conseiller de vous y intéresser.

④Niels Høeg
Skakbladet, 1923



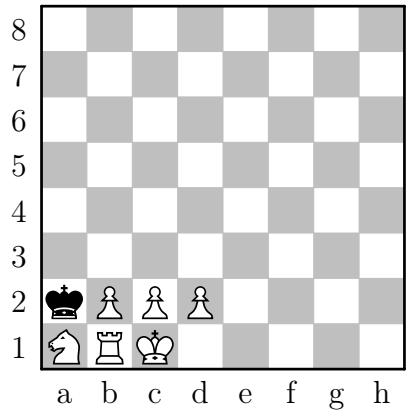
Dernier coup?

⑤Niels Høeg
Skakbladet, 1923



Dernier coup?

⑥Vilhelm Røpke
Skakbladet, 08/1924



Dernier coup?

Le premier problème d'analyse rétrograde connu est une composition échiquéenne de Max Lange datée de 1858 (avec une clé de prise en passant, coup spécial du jeu d'Échecs et premier effet rétro analysé). Bien sûr pas d'informatique à cette époque, juste l'habileté des compositeurs de problèmes, dont l'incroyable Samuel Loyd. Il existe maintenant plusieurs dizaines de milliers de problèmes d'analyse rétrograde aux Échecs.

Le problème ④ de Niels Høeg est le plus simple possible. Les noirs sont en échec, donc les blancs viennent de jouer le pion qui donne échec, et la seule case de provenance de ce pion est h4. Le dernier coup est donc h4-h5+.

Notons dans le ④ que si les pièces étaient toutes un cran plus bas, la possibilité pour le pion d'effectuer un double pas à partir de sa ligne d'origine n'aurait pas permis de conclure, car h3-h4 et h2-h4 permettent tous deux d'arriver en h4. Je pense que ce sont les règles un peu particulières du jeu d'Échecs qui permettent autant de richesse dans les problèmes, directs ou rétrogrades, et nous allons tout de suite voir une autre de ces règles spéciales en action.

Les noirs sont de nouveau en échec dans le problème ⑤, cette fois par le fou. Mais le fou ne pouvant pas sauter par dessus une autre pièce, il n'a pas de case de provenance possible. Il a donc dû arriver en c8 d'une autre manière... Et sa position en 8ème rangée, ainsi que sa couleur blanche sont des indices d'un autre coup spécial des Échecs: une promotion! Et plus précisément une sous-promotion: un pion était en c7, s'est déplacé en c8 est s'est promu en fou qui donne échec, ce que l'on note par c7-c8=F+.

Dans le problème ⑥ de Vilhelm Røpke (voir 1924-08.pdf) on voit que Noir n'a pas pu jouer en dernier, car son roi n'a pas de case de provenance valide (en a3 ou b3 il aurait été en échec par des pions qui n'ont pas bougé). C'est donc Blanc qui a bougé en dernier, seuls son cavalier et son roi ont pu bouger. Supposons que le cavalier blanc ait bougé de b3 vers a1 sans capturer: bien que cela ne soit pas demandé, interrogeons nous sur quel était alors le coup noir précédent? La seule possibilité est alors le roi qui est allé de a1 vers a2 (il ne peut toujours pas venir de a3 ou b3). Mais en reprenant ce coup, le roi noir en a1 aurait été en échec illégal par la tour blanche qui n'a pas pu bouger pour donner cet échec (il est aussi en échec par le cavalier, mais ce n'est pas cela qui pose problème car lui aurait pu bouger, par exemple depuis c5 et donner échec). Nous avons une impossibilité, notre hypothèse est donc fausse. Supposons que le le cavalier blanc ait bougé de b3 vers a1 en ayant capturé une pièce noire. Mais alors les noirs n'ont toujours aucun coup possible, car quelle que soit la pièce noire que l'on place en a1, elle ne peut pas bouger et le roi noir non plus. Nous avons une impossibilité, notre hypothèse est donc fausse. Donc c'est le roi blanc qui a bougé en dernier.
Supposons que le roi blanc ait bougé de d1 vers c1 sans capturer. Comme précédemment le roi noir ne peut pas avoir de coup de provenance possible. Nous avons une impossibilité, notre hypothèse est donc fausse.
Donc le roi blanc a bougé de d1 vers c1 en capturant une pièce noire. Laquelle? On se convainc rapidement que seul un cavalier noir permet de donner un coup antérieur aux noirs.
Donc le dernier coup joué est roi blanc joue à partir de d1 et capture un cavalier noir qui était en c1, ce que l'on note Rd1×Cc1.

On le voit, les solutions d'un problème d'analyse rétrograde sont plus longues que celles d'un problème direct, mais elles sont aussi plus mathématiques. Le livre "Sherlock Holmes en Échecs" de Raymond Smullyan (malheureusement épuisé), est une bonne initiation au genre malgré quelques erreurs typographiques dans la version Française.

Petit historique de l'algorithme d'induction rétrograde

Des algorithmes pour jouer aux Échecs sont conçus peu après l'apparition des premiers ordinateurs, par exemple Turochamp par Alan Turing et David Champernowne en 1948, bien qu'il n'ait jamais été implémenté. Les premiers programmes fonctionnels apparaissent dans les années 1950-1960, avec un fort engouement pour cette recherche à partir de la fin des années 1960. Leurs algorithmes utilisent une recherche en avant dans l'arbre de jeu en utilisant rapidement la méthode du minimax puis ses améliorations telles que l'élagage alpha-bêta... Les différences des programmes résidaient alors essentiellement dans les routines d'évaluation de position et dans leurs librairies d'ouvertures.

Les développements pour le jeu d'Échecs se transposent facilement pour les autres jeux combinatoires (Dames, Othello, Go...) mais les efforts sont surtout portés sur les Échecs dans un premier temps, avec deux jalons pour le grand public: en 1988, HiTech est le premier programme qui bat un grand-maître d'Échecs, puis Deep Blue bat le champion du monde Garry Kasparov en 1997.

Parallèlement à ces recherches pour le jeu, Richard Bellman propose en 1965 une méthode pour créer par analyse rétrograde une base de donnée pour les positions du jeu d'Échecs et du jeu de dames anglaises. Il s'agit de partir des positions terminales (mat ou pat pour le jeu d'Échecs) et de regarder comment les autres positions permettent d'aboutir à ces positions terminales.

Dans les années 1970, certaines finales du jeu d'Échecs sont entièrement analysées par analyse rétrograde, notamment par Ken Thompson et ses collègues qui ont mis dès que possible leurs bases de données à disposition, avec l'avènement du CD-ROM, à cause de la taille des fichiers (par exemple la finale roi+tour+pion contre roi+tour occupe 60 méga-octets). En 2018 toutes les finales avec 7 pièces sont explorées, leur stockage demande plusieurs terra-octets.

Depuis les années 2000, les bases de données créées par analyse rétrograde sont utilisées par les programmes de jeu pour leur permettre de jouer parfaitement les finales dès lors que le nombre de pièces en présence diminue, la recherche en avant est renforcée par les bases de données issues d'une recherche en arrière.

Ce duo recherche en avant et recherche en arrière a aussi permis à Jonathan Shaeffer et son équipe de résoudre le jeu de dames anglaises en 2014: si les deux joueurs jouent au mieux, le résultat sera une partie nulle.

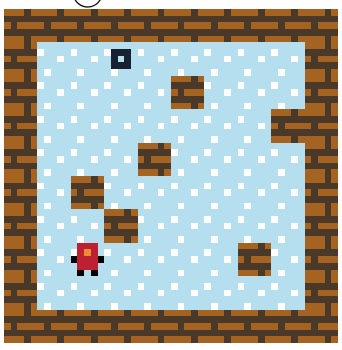
Dernier point clé dans ce petit historique autour des jeux combinatoires, en 2016 le programme Alpha Go bat Lee Seedol, un des meilleurs joueurs mondiaux de Go. Ce résultat fut plus surprenant que les précédents jalons tellement le jeu de Go semblait hors de portée des programmes informatiques à cause du nombre beaucoup plus grand de possibilités, et aussi car il a marqué un tournant dans les algorithmes utilisés par les programmes de jeu, puisqu'il utilisait des réseaux de neurones couplé à de l'apprentissage automatique. Cette technique a été adaptée avec succès à d'autres jeux: Alpha Zéro pour les Échecs et le Shogi, Alpha Star pour le jeu vidéo StarCraft...

L'algorithme d'induction rétrograde reste toutefois utile, car contrairement à ces nouveaux algorithmes, il ne se contente pas d'être très fort, mais il garantit des lignes de jeu parfaites lorsqu'il y a peu de pièces.

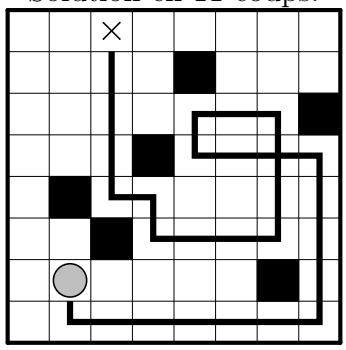
Illustration de l'algorithme d'induction rétrograde

Dans le jeu Slidyyyyyyy de Moskemoe ⑦ , le petit robot doit s'arrêter sur le carré noir, mais il se déplace en ligne droite et ne s'arrête que quand il rencontre un mur.

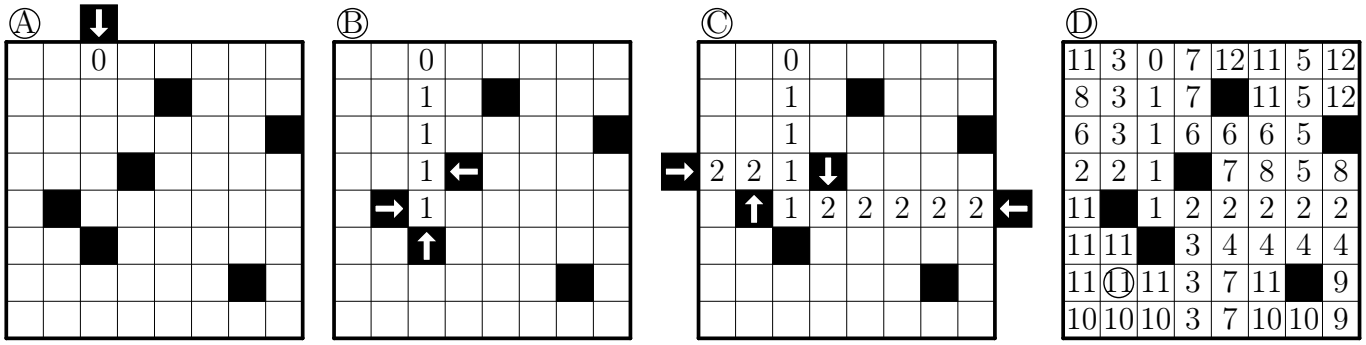
⑦Moskemoe



Solution en 11 coups.



La solution demande 11 coups et pour la trouver nous allons étudier toutes les positions possibles du robot sur la grille (les états possibles) en commençant par la ou les positions terminales.



Lorsque le robot est sur le carré noir, nous indiquons que nous sommes à la distance 0 de la solution en inscrivant un 0 pour cette position. Puisque le robot a dû s'arrêter sur cette case, il doit venir de la direction indiquée par la flèche Ⓐ. Sur toute cette ligne nous sommes alors à une distance de 1 de la solution, ce que nous indiquons en inscrivant un 1 pour ces positions. Il y a alors 3 manières de s'être arrêté sur un de ces positions marquées 1 en touchant un mur, comme montré par les flèches Ⓑ. Sur toutes les lignes indiquées par ces 3 flèches, nous indiquons que nous sommes à une distance de 2 de la solution, sauf si nous avions déjà une valeur indiquée. Et la encore des flèches indiquent les 4 manières de s'être arrêté sur une de ces positions Ⓒ. En continuant ainsi le processus jusqu'à ce que toutes les positions soient explorées (ou dans certains cas jusqu'à ce que nous n'arrivions plus à progresser) nous obtenons une base de données des distances à la solution Ⓓ.

La position initiale du robot est entourée, et se trouve à une distance de 11. A partir de la on va vers la position qui se trouve à une distance de 10, puis à celle qui se trouve à une distance de 9... Cela nous donne le chemin vers la solution.

C'est ce principe qui a été utilisé lors de la création de problèmes pour différents jeux: mon jeu informatique Slidings, les défis de "Parc'Ours en forêt" et de "Chats tournent en rond", deux jeux édités par Smart Games et inventés par Raf Peeters avec mon aide pour la création informatique des défis.

Dans l'algorithme que j'ai utilisé pour créer les bases de données permettant ensuite de créer des problèmes, j'utilise un tableau DTS (Distance To Solution) qui contient le statut de chaque position possible du jeu: inconnue, illégale ou le nombre de coups qui la sépare du gain.

Ce qui est désigné par le terme position va varier d'un jeu à l'autre. Pour Slidyyyyyyy il s'agit juste des 64 positions possibles du robot sur le plateau (si le plateau contient un mur la position est marquée comme illégale). Pour slidings qui partage à peu près le même principe, mais sur un plateau de N cases et K billes, il y aura alors $N \times (N-1) \times ... \times (N-K+1)$ positions possibles des K billes.
Pour le morpion on prendra plutôt en compte toutes les grilles possibles, chacune des 9 cases pouvant être vide, contenir une croix ou un rond, il y a aura $3^9 = 19683$ positions possibles, dont un grand nombre sont en fait illégales, par exemple une position avec 2 croix et aucun rond.
Pour le jeu "Parc'Ours en forêt" une position était donnée par l'orientation de chacun des 9 arbres (dont le type était donné au préalable) et la position de chacune des pièces de jeu. Ce fut un jeu difficile à programmer à cause de la complexité des mouvements: rotation des arbres qui entrainent les pièces dans un mouvement non linéaire.
Pour le jeu "Chats tournent en rond" toutes les positions possibles des polyminos troués étaient générées au préalable en indiquant celles qui sont "proches" les unes des autres (un seul polymino bouge), et numérotées ce qui permettait d'y faire référence avec ce numéro seul.
Pour les jeux à plusieurs joueurs il faut aussi indiquer qui a le trait, prendre en compte la possibilité d'un match nul, et pour un jeu comme les Échecs la définition d'une position peut être encore bien plus complexe: il faut indiquer si les roques ou des prises en passant sont possibles, se souvenir des coups précédents pour prendre en compte la règle de répétition à trois reprise de la position...

Voici l'algorithme:

Passe n°0:
Pour chaque état de jeu:
▷ Si la position est illégale, l'indiquer en mettant son DTS à -99
▷ Si la position est gagnante, l'indiquer en mettant son DTS à 0
▷ Les autres positions sont marquées comme inconnues en mettant le DTS à -1

Passe n°N, tant que le nombre de positions dont la DTS est inconnue (-1) décroit:
Pour chaque position avec une DTS inconnue, si elle permet d'aller par un coup légal à une position avec une DTS connue et positive (en général N-1) alors mettre son DTS à N.

Lorsque toutes les positions ont un DTS connu ou qu'il n'est plus possible de progresser, notre base de donnée DTS est complète. Les éventuelles positions qui gardent une DTS inconnue (-1) ne permettent pas d'atteindre une position gagnante.

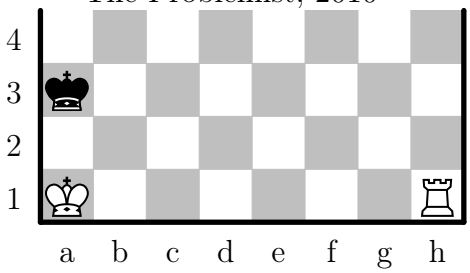
Dans les jeux pré-cités, je sauvais aussi un lien entre la position à une distance N et sa première prédecesseure trouvée à une distance $N-1$, ce qui me permettait ensuite d'exhiber rapidement une solution complète. Mais ce ne sera pas toujours satisfaisant, car la solution ainsi obtenue ne sera peut-être pas la plus logique, par exemple elle pourrait faire bouger la pièce A, puis la pièce B, puis de nouveau la pièce A alors que bouger deux fois la pièce A puis la pièce B est possible. Pour améliorer cela il faut soit sauver tous les liens vers les positions prédecesseures, mais leur nombre est variable, soit faire une recherche en avant ce qui demande un autre algorithme.

Dans l'exemple du Morpion ou de Slidyyyyyyy la base de donnée est particulièrement petite, mais ce sont des cas exceptionnels, il faut plutôt compter des millions d'entrées ou plus. Un travail important d'analyse de la base de donnée est alors à faire pour découvrir des positions intéressantes en elles mêmes ou intéressantes à cause de leurs solutions.
On peut rechercher par exemple des coups forcés, une pièce qui effectue un aller retour pour en laisser passer une autre, une pièce qui effectue un circuit, une disposition particulière des pièces, la position avec la plus longue solution (aux Échecs le record actuel est un mat en 584 coups avec 8 pièces trouvé par Marc Bourzutschky), etc...

Le tango: en avant, en arrière, ou la complémentarité des deux approches

L'analyse rétrograde est aussi utilisée pour la création de problèmes directs comme dans le problème ⑧ ou dans des études de fin de partie qui utilisent les bases de finales (l'utilisation de résultats informatique étant d'ailleurs un sujet récurrent de controverses). C'est un peu la situation inverse du problème ③ qui était un problème rétrograde créé avec un algorithme d'exploration en avant, ce qui montre encore que les deux approches s'enrichissent mutuellement.

⑧AB et Stephen Emmerson
Dédié à Éric Angelini
The Problemist, 2010



a) Blanc enlève une case et mate deux fois plus vite.
b) Blanc enlève deux cases et mate huit fois plus vite.
c) Noir enlève deux cases et annule.

Dans ce problème d'échecs féerique, nous demandons d'interdire l'accès à certaines cases du plateau. Il a été conçu avec un programme maison d'analyse rétrograde dans lequel il était possible de modifier la forme du plateau.

a) Blanc peut mater en 8 coups avec la séquence forcée suivante: 1.Tb1 Ra4 2.Ra2 Ra5 3.Ra3 Ra6 4.Ra4 Ra7 5.Ra5 Ra8 6.Rb6 Rb8 7.Tc1 Ra8 8.Tc8#. Mais si Blanc enlève la case c3, il peut alors mater en 4 coups: 1.Tg4 Rb3 2.Rb1 Ra3 3.Rc2 Ra2 4.Ta4#.
b) Si Blanc enlève les cases a4 et b4, il mate par 1.Th3#.
c) Si Noir enlève g1 et h2 la tour blanche ne peut plus bouger.

Même si nous ne pouvons pas voyager dans le temps comme les personnages de science fiction, j'espère vous avoir montré que l'analyse rétrograde permet d'explorer le passé avec une bonne dose de logique, avec parfois des calculs informatiques, mais surtout avec beaucoup de plaisir pour les amateurs de réflexion.

Les liens en clair:

- http://abrobecker.free.fr/ : mon site web.
- http://abrobecker.free.fr/text/tictactoe.pdf : analyse rétrograde et tic-tac-toe.
- http://abrobecker.free.fr/text/othello.pdf : analyse rétrograde et Othello.
- https://erich-friedman.github.io/ : site web d'Erich Friedman.
- https://joekisenwether.wordpress.com/non-chess-retrograde-analysis/ : page de Joseph Kisenwether sur l'analyse rétrograde dans les jeux autres que le jeu d'Échecs.
- https://pdb.dieschwalbe.de/search.jsp?expression=PROBID='P0001360' : problème de Max Lange.
- https://pdb.dieschwalbe.de/search.jsp?expression=A='Loyd'+AND+G='retro' : problèmes de Samuel Loyd.
- https://skak.dk/images/skakbladet/1924/1924-08.pdf : le magazine Skakbladet.
- https://pdb.dieschwalbe.de/search.jsp?expression=A='Hoeg'+AND+G='retro' : problèmes de Niels Høeg.
- https://pdb.dieschwalbe.de/search.jsp?expression=A='Röpke'+AND+G='retro' : problèmes de Vilhelm Røpke.
- https://fr.wikipedia.org/wiki/Turochamp : Turochamp par Alan Turing et David Champernowne.
- https://fr.wikipedia.org/wiki/HiTech : HiTech par Hans Berliner et son équipe.
- https://fr.wikipedia.org/wiki/Matchs_Deep_Blue_contre_Kasparov#Match_revanche_(1997) : Deep Blue bat le champion du monde Garry Kasparov.
- https://www.researchgate.net/publication/231216842_Checkers_Is_Solved : article de Jonathan Shaeffer et son équipe qui ont résolu le jeu de dames anglaises.
- https://fr.wikipedia.org/wiki/Match_AlphaGo_-_Lee_Sedol : Alpha Go bat Lee Seedol.
- https://www.puzzlescript.net/play.html?p=9898418 : le jeu Slidyyyyyyy de Moskemoe.
- https://www.puzzlescript.net/play.html?p=8931824 : mon jeu Slidings.
- https://www.smartgames.eu/fr/jeux-pour-1-joueur/ : le jeu "Parc'Ours en forêt".
- https://www.smartgames.eu/fr/jeux-pour-1-joueur/chats-tournent-en-rond : le jeu "Chats tournent en rond".
- https://www.smartgamesandpuzzles.com/single-player-games.html : le site web de Raf Peeters.
- https://wismuth.com/chess/statistics-positions.html : le site web de François Labelle.
- https://en.chessbase.com/post/8-piece-endgame-tablebases-first-findings-and-interview : une interview de Marc Bourzutschky.
- https://www.arves.org/arves/images/PDF/EG_PDF/eg173.pdf : discussions sur l'usage des bases de finales dans le magazine EG.