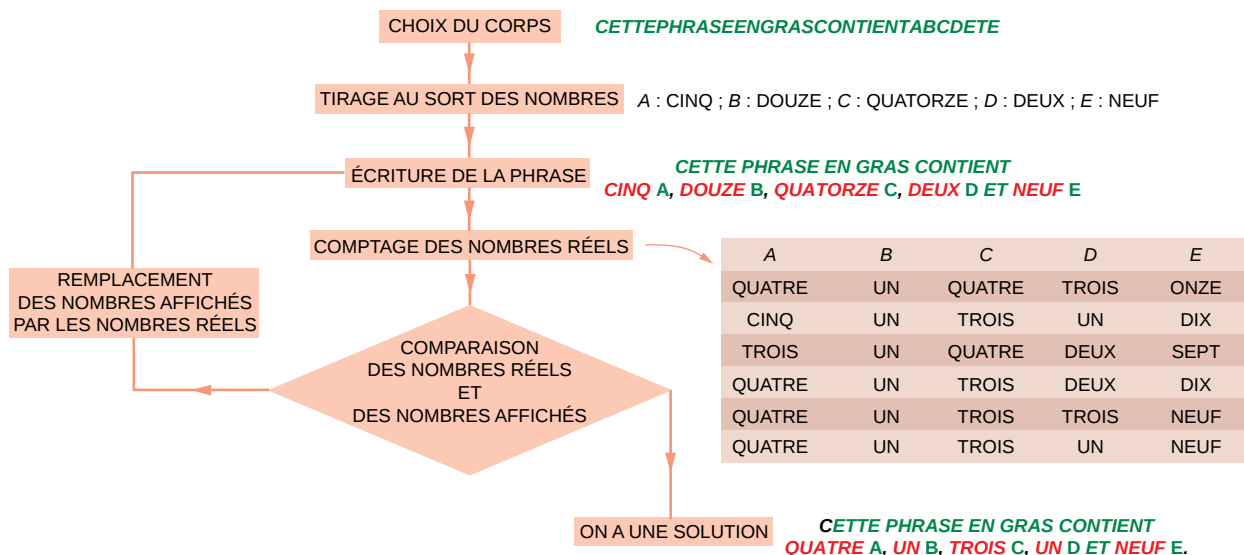




Un algorithme permet de trouver des phrases réflexives dont on a initialement choisi le corps : on tire au sort les nombres associés aux lettres, on écrit la phrase, et on compare ces nombres affichés avec les nombres réels. Si des nombres réels sont différents des nombres affichés, il les remplace, et on recommence jusqu'à ce que l'on ait trouvé une solution. Dans l'exemple, volontairement limité aux cinq premières lettres de l'alphabet, on obtient une solution après six itérations seulement (les

valeurs des nombres réels à chaque itération sont donnés dans le tableau à droite). Avec tout l'alphabet, plusieurs milliers d'itérations sont généralement nécessaires, et l'on ne trouve pas nécessairement de solution, même s'il en existe une. Une amélioration de la méthode consiste à ne remplacer, à chaque étape, qu'un seul des nombres affichés par le nombre réel de la lettre correspondante : on bouleverse ainsi moins la phrase à chaque étape, et on trouve plus fréquemment une solution.

Par cette méthode, nous trouvons quelquefois des solutions. Son succès est surprenant, car la phrase est bouleversée à chaque itération : la valeur de la plupart des nombres appartenant à la partie mobile est modifiée à chaque étape. Chaque phrase est donc très différente de la précédente, et l'on ne s'approche pas régulièrement de la solution (si elle existe) : on peut successivement s'en approcher et s'en éloigner.

Toutefois, ce procédé a un avantage important : si l'on arrive à une situation où l'ensemble des lettres des nombres réels est identique à l'ensemble des lettres des nombres affichés, on obtient la solution à l'étape suivante. Supposons que nous ayons dans la phrase du titre et du chapeau « quatre t », au lieu de vingt-quatre, et « vingt-sept x » au lieu de sept, les autres nombres étant les mêmes. Les nombres réels donneront, à l'étape suivante, « vingt-quatre t » et « sept x », ce qui est la solution.

### UN SEUL REMPLACEMENT

J'ai amélioré cette méthode. Au lieu de remplacer à chaque itération tous les nombres affichés par les nombres réels correspondants, le programme choisit une seule lettre dont le nombre affiché est différent du nombre réel, et il ne remplace que le nombre affiché de cette lettre par son nombre réel. Cette nouvelle méthode doit préserver une qualité essentielle de la précédente : quand l'ensemble des lettres des nombres réels est le même

que l'ensemble des lettres des nombres affichés, on doit obtenir la solution à l'étape suivante. Sinon, on risque de passer à côté d'une solution évidente. J'ai donc ajouté une vérification : le programme détermine l'ensemble des lettres des nombres réels et celui des lettres des nombres affichés qui en diffèrent. Si ces deux ensembles sont identiques, l'ensemble des nombres réels est une solution. Pour ne pas trop allonger le temps de calcul, le programme ne fait cette comparaison que si l'écart entre les nombres réels et les nombres affichés (l'écart est la somme des valeurs absolues des différences entre les nombres réels et les nombres affichés) n'est pas trop grand.

Cette amélioration est performante : elle permet de trouver plus de solutions. Cela est dû à ce que la phrase est moins bouleversée à chaque itération, puisque l'on ne change qu'un seul nombre affiché : la partie mobile est plus stable.

Pour trouver de nombreuses phrases, j'ai introduit dans le programme un générateur de corps de phrases. J'ai défini d'une part un « noyau », suite de mots commune à tous les corps, et j'ai défini d'autre part plusieurs endroits, dans le noyau, où le programme peut insérer un « ajout », groupe de mots pris parmi plusieurs possibilités liées à chaque endroit. La présence ou l'absence de ces ajouts ne change pas le sens du corps : par exemple, « cher lecteur », « chers lecteurs », « ami lecteur » en tête de phrase ou « pour terminer », « finalement », « pour finir » avant le z final. En combinant une dizaine de

possibilités d'ajout, on crée de nombreux corps. À partir d'un même noyau, le programme engendrait 12 288 corps différents, ce qui pouvait donner naissance à environ 9 000 phrases réflexives.

### PLURALITÉ DE SOLUTIONS

Le même corps donne parfois plusieurs solutions. Pour le cas de deux caractères, nous avons vu la phrase : « Cette phrase contient exactement cinq n et trois o. » Il existe d'autres solutions sur le même corps, par exemple : « Cette phrase contient exactement quatre n et trois o. », « Cette phrase contient exactement cinq n et deux o. », « Cette phrase contient exactement quatre n et deux o. »

Ce phénomène est fréquent, même quand on indique les nombres d'apparitions des 26 lettres de l'alphabet : pour les corps ayant au moins une solution, le nombre moyen de solutions est d'environ 1,5. Le record actuel est de dix phrases différentes construites sur le même corps. Voici l'une de ces dix phrases : « Vous avez bien raison de compter toutes les lettres de cette phrase avant de la croire quand elle vous affirme qu'elle contient quinze a, deux b, sept c, sept d, quarante-trois e, quatre f, cinq g, trois h, vingt et un i, un j, un k, huit l, quatre m, vingt-cinq n, treize o, sept p, douze q, vingt et un r, quinze s, trente-quatre t, vingt-quatre u, neuf v, un w, trois x, un y et pour terminer six z. »

À partir d'un point de départ, on peut soit arriver à une solution, soit tourner indéfiniment dans une boucle.

Cette dernière situation est la plus fréquente, d'une part parce qu'il y a des corps sans solution et d'autre part parce qu'il arrive souvent qu'un point de départ conduise à une boucle même quand il existe une solution. La phrase sans e n'a pas été trouvée une seule fois par le programme, bien qu'il ait essayé 200 000 points de départ.

Il arrive aussi que l'on ne trouve pas de solution parce que l'on n'a pas poussé assez loin les itérations : certaines solutions n'apparaissent qu'après plus de 10 000 itérations. Dans la plupart des cas, on les trouve toutefois avant 5 000 itérations, et souvent avant 2 000. Par ailleurs, une solution obtenue après beaucoup d'itérations pour un point de départ peut l'être avec bien moins d'itérations pour un autre point de départ. Il est donc préférable d'augmenter le nombre de points de départ et de diminuer le nombre d'itérations.

Avec des valeurs assez grandes pour les nombres de points de départ et d'itérations, le programme trouve au moins une solution pour plus de la moitié des corps. Le nombre total de solutions est encore plus élevé à cause des solutions multiples : il vaut environ trois quarts du nombre de corps essayés.

Ces méthodes réussissent parce que, l'expérience le montre, la probabilité est grande qu'il existe une solution pour un corps quelconque. Elles sont toutefois inefficaces pour trouver une phrase sans e, parce qu'il est très improbable qu'un corps conduise à une telle solution : on ne peut utiliser qu'à peine la moitié des nombres inférieurs à 30 (et aucun au-delà), et la probabilité que tous les 18 nombres cherchés soient dans cet ensemble est faible. Aussi ai-je construit sans l'aide de l'ordinateur la phrase sans e citée plus haut. J'ai d'abord choisi les nombres affichés pour chaque lettre. En enlevant les lettres de ces nombres de l'ensemble des lettres décrit par ces mêmes nombres, j'ai obtenu l'ensemble des lettres du corps : si je décide que la phrase contiendra cinq q, j'enlève une unité pour le q qui est dans cinq que je viens d'affecter à q ; le corps de la phrase contiendra quatre q (dont la lettre isolée). J'enlève aussi un des quantités affectées aux lettres c, i et n. Ayant ainsi obtenu les lettres du corps, j'ai recherché un anagramme constituant une phrase correcte au point de vue de la syntaxe et du sens. Mon choix initial des nombres affichés a été guidé par les fréquences des lettres en français, afin de maximiser mes chances de résoudre ensuite l'anagramme.

«Au fond, est-ce bien utile de savoir que cette phrase contient cinq a, deux b, huit c, huit d, vingt e, quatre f, cinq g,

six h, vingt-deux i, un j, un k, deux l, un m, vingt-deux n, cinq o, trois p, sept q, cinq r, huit s, dix-neuf t, vingt u, six v, un w, huit x, un y et enfin un z?» De nombreux problèmes du monde réel, à cause de liens très étroits entre différents domaines, ont les mêmes caractéristiques que la construction de phrases réflexives : chaque action a des répercussions importantes sur la situation et détruit ce qui était satisfaisant.

L'évolution des espèces, par exemple, est mue par un tel mécanisme, selon l'hypothèse de la Reine Rouge proposée par Leigh Van Valen : si l'une des espèces qui exploitent un milieu évolue dans le sens d'une meilleure adaptation, accroissant ainsi son succès reproductif, elle diminue la qualité de l'adaptation des autres espèces. Ces dernières évoluent alors pour survivre, annulant ainsi l'avantage de la première. De même, la protection d'un carnivore devenu rare dans un milieu naturel a des répercussions sur la survie de ses proies, qui peuvent à leur tour se raréfier exagérément. Les hommes politiques veulent tous relancer l'économie et réduire le chômage : ils prennent une mesure qui paraît raisonnable pour le but cherché, mais qui a des conséquences imprévues et néfastes sur d'autres points ; cela les oblige à prendre une autre mesure pour y remédier, mais cette nouvelle mesure a des conséquences négatives pour le but initial. Il serait intéressant de trouver des méthodes efficaces pour résoudre de tels problèmes.

En outre, la méthode utilisée donne de bien meilleurs résultats que ce que nous attendions. Capte-t-elle une régularité encore inconnue du problème, voire du monde ? Y a-t-il d'autres problèmes pour lesquelles une méthode analogue obtient des succès contre toute attente ? Le succès est-il dû à l'utilisation d'un mécanisme évolutif, où l'on corrige systématiquement les erreurs décelées sans se préoccuper des conséquences souvent néfastes de ces corrections ? Nous sommes surpris, et il est toujours utile de réfléchir pour comprendre les raisons d'une surprise.

---

**Jacques PITRAT est directeur de recherche CNRS au Laboratoire d'informatique de l'Université Paris 6.**

**Jacques PITRAT, *Ce titre contient quatre a, un b, cinq c...*, rapport de recherche 96 / 26 du LAFORIA (Paris 6). Ce rapport contient le listing du programme C qui a créé ces phrases réflexives. Il est téléchargeable à l'adresse <http://www.apa.lip6.fr/META/pitrat.html>**

**Douglas HOFSTADTER, *Ma Thémagie*, InterEditions, 1988.**

---