



HAL
open science

Le meilleur des deux mondes

Christine Solnon

► **To cite this version:**

| Christine Solnon. Le meilleur des deux mondes. La Recherche, 2023, pp.1-2. hal-04209801

HAL Id: hal-04209801

<https://hal.science/hal-04209801>

Submitted on 2 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Le meilleur des deux mondes

Christine Solnon, CITI, Inria, INSA Lyon, F-69621 Villeurbanne

Chronique parue dans La recherche N° 575, oct-déc 2023

En 1972, à Marseille, Alain Colmerauer et son équipe de recherche mettaient au point Prolog, un langage de programmation en logique. Celui-ci est devenu célèbre dix ans plus tard lors du lancement du projet japonais Cinquième génération, visant à créer une nouvelle architecture consacrée à l'intelligence artificielle (IA) et fondée sur Prolog. C'était l'âge d'or des systèmes experts, IA symboliques qui reproduisent le raisonnement d'experts humains pour résoudre des problèmes. Si Prolog a marqué l'histoire de l'informatique, c'est parce qu'il est radicalement différent des langages de programmation classiques, dits *impératifs*, tels que C, Java ou Python. Dans ces derniers, un programme est une succession d'instructions qui modifient les valeurs contenues dans la mémoire. En Prolog, un programme est un ensemble de formules logiques qui définissent les connaissances sur un sujet donné. Pour exécuter le programme, on lui pose des questions auxquelles un moteur d'inférence (un algorithme effectuant des raisonnements déductifs) répond en évaluant s'il s'agit de conséquences logiques du programme.

Cinquante ans après sa création, quel est l'héritage de Prolog ? Dans les années 1990, ce langage a été étendu de sorte à intégrer des contraintes aux formules logiques : par exemple, pour imposer que la somme d'un ensemble de variables soit égale à une valeur donnée, ou qu'un ensemble de variables prennent des valeurs toutes différentes deux à deux. Le mécanisme d'inférence logique a alors été remplacé par un mécanisme plus général de résolution de systèmes de contraintes, appelé programmation par contraintes. Afin de résoudre un problème avec cette approche, il suffit de le décrire en termes de variables (les inconnues du problème) et de contraintes sur ces variables (les relations entre les inconnues) ; la solution est alors cherchée automatiquement par un moteur de résolution intégré. Il existe des solveurs commerciaux (IBM ILOG CP Optimizer, par exemple), ainsi que des bibliothèques open source (telles que Choco en Java, ou PyCSP3 en Python), et ces outils sont très largement utilisés dans l'industrie en vue de résoudre

des problèmes combinatoires tels que des problèmes d'ordonnancement ou d'emploi du temps.

Prolog a été initialement conçu pour le traitement automatique des langues naturelles. À l'ère de ChatGPT, il est intéressant de comparer ces deux approches radicalement opposées. Le langage Prolog procède d'une approche symbolique fondée sur la logique : un expert humain conçoit un modèle logique définissant les règles de composition des phrases, et ce modèle est ensuite utilisé par un moteur d'inférence pour analyser ou générer un texte. Cette approche permet de raisonner et aussi d'explicitier les raisonnements qui ont été faits à un être humain. Malheureusement, les langues naturelles sont trop complexes pour pouvoir être totalement modélisées par des règles logiques. De plus, les règles définies par les experts supposent que les textes ne comportent pas de fautes ni de nouveaux mots, ce qui est rarement le cas en pratique.

À partir des années 1990, les approches symboliques pour le traitement automatique des langues ont été remplacées par des approches statistiques, qui ont récemment débouché sur les *grands modèles de langage* tels que ChatGPT. Avec ces modèles, une grande quantité de textes sont utilisés pour apprendre à prédire le mot le plus probable suivant un début de texte donné, sans qu'aucune règle ne soit explicitement construite. Grâce à cette approche statistique, on produit des textes remarquables, à la fois bien construits et pertinents, mais sans raisonnement ni compréhension.

Prenons un cas de figure : nous demandons à ChatGPT si la somme de trois entiers impairs peut être paire, il nous répond :

Oui, (...) il est possible de choisir des entiers impairs de manière à obtenir une somme qui soit un nombre pair.

Par exemple : $1 + 3 + 4 = 8$ (somme de trois entiers impairs, résultat pair).

Cet exemple montre que, s'il a bien identifié le sujet de la question, ChatGPT ne comprend pas ce qu'il écrit (4 n'est pas un entier impair). Pour y remédier, une piste actuellement étudiée consiste à combiner les approches symbolique et statistique en s'inspirant de travaux en psychologie décrits notamment par le psychologue et économiste Daniel Kahneman (prix Nobel d'économie 2002), et montrant que le cerveau humain combine deux modes de pensée : un système rapide qui fonctionne de manière automatique et inconsciente, et un système lent qui fait appel à la logique et permet de résoudre des problèmes complexes. En associant ainsi le meilleur des deux mondes (logique et statistique) on pourrait peut-être aboutir à des IA qui n'hallucinent plus.