É. GILLET

L'ordinateur : un outil au service du logicien

Les cahiers de l'analyse des données, tome 15, nº 3 (1990), p. 323-330

<http://www.numdam.org/item?id=CAD_1990__15_3_323_0>

Les Cahiers de l'Analyse des Données
Vol XV - 1990 n° 3 - pp. 323-330.

L'ORDINATEUR: UN OUTIL AU SERVICE DU LOGICIEN

[ORD. LOG.]

*É. GILLET**

1 Introduction

L'apparition de l'ordinateur à tous les échelons de la recherche scientifique n'est pas sans en avoir modifié, parfois considérablement, les méthodes de travail. La philosophie n'échappe pas à cette évolution, mais y présente un paysage particulier.

La philosophie, depuis ses origines, ne s'est pas contentée de développer ses propres outils, elle les a incorporés parmi ses objets d'étude. Le cas de la logique, attelée à cette double tâche, est symptomatique du rapport particulier de la philosophie à l'ordinateur.

En plus des utilisations bureautiques de l'ordinateur (traitement de texte, mise en page, gestion de fichiers, planification, etc.), les logiciens ont développé leurs propres outils : démonstrateurs automatiques, systèmes d'aide interactive à la démonstration, bases de connaissances et même des langages de programmation comme PROLOG. Ceci constitue la première tâche: l'utilisation des ressources informatiques dans le cadre de la logique. Déjà ici, on observe les répercussions de l'introduction de ces outils nouveaux sur les sciences de l'ordinateur. En effet la logique formelle fait partie intégrante des bases théoriques de l'informatique tant du point de vue des grammaires formelles que de la théorie de la calculabilité. Mais si l'interaction entre les deux disciplines est assez directe, elle ne diffère pas en nature du choc en retour de la pratique sur la théorie qui s'observe dans toutes les sciences jeunes.

Ce qui va constituer le cœur de notre intervention d'aujourd'hui dépasse le cadre de cette première tâche et concerne la seconde : le développement des outils informatiques comme objet étudié par la logique. Nous nous limiterons pour la circonstance à mentionner le problème des démonstrateurs automatiques pour illustrer le changement de critère induit par la théorie de la complexité, puis

(*) Université de LIÈGE; Séminaire de logique et d'épistémologie; 32, place du XX Août, B 4000 LIÈGE.

Les cahiers de l'analyse des données - 0339 - 3097/90/ 02 169 12/ $ 3.20/©Gauthier-Villars

nous nous appuierons sur l'exemple des bases de connaissances pour esquisser les perspectives nouvelles qui s'offrent à notre regard.

2 Logique et complexité

Dans les années 30, la logique a connu des progrès fulgurants. GÖDEL établit une première partition d'importance entre les théories complètes et incomplètes. Ainsi certaines propositions vraies de l'arithmétique s'avèrent impossibles à démontrer quel que soit le système axiomatique choisi. Parallèlement à cette découverte se pointe une autre partition liée à la notion de calculabilité ou décidabilité. La thèse de CHURCH est que toute fonction intuitivement calculable est récursive. Ainsi la notion intuitive de décidabilité, liée à l'existence d'une procédure de décision effective, se ramène-t-elle à la notion formelle de récursivité. D'autre part on prouve l'équivalence entre la classe des fonctions récursives et la classe des problèmes calculables par un automate abstrait appelé machine de TURING.

L'apparition puis la banalisation des ordinateurs, qui matérialisent la machine de TURING, - un ordinateur est une machine universelle à ruban fini - vont rendre accessible un nouveau champ de recherche et de réalisations: de théorique, la notion de démonstrateur automatique va devenir pratique. Il sera désormais possible d'envisager une machine qui prouve un théorème ou qui vérifie la validité d'une phrase logique. Ainsi, la détermination de la validité d'un énoncé du calcul des propositions se révèle décidable et donc soluble par une machine de TURING ; la vérification de la validité d'un énoncé contenant des prédicats, quant à elle, n'est que semi-décidable. Un ordinateur chargé de ce problème répondra oui si la phrase est valide, il tournera indéfiniment si la phrase est simplement consistante.

Malheureusement, la distinction entre un problème décidable, indécidable et semi-décidable est en pratique inutilisable. Si, dans ce dernier cas, on sait que pour une phrase valide l'ordinateur s'arrêtera avec la bonne réponse, on ne sait pas quand il s'arrêtera et, pour peu que le calcul soit long, on ne pourra pas déterminer si la machine boucle sur une phrase simplement consistante ou si, examinant une phrase valide, elle s'approche de la solution. Il est donc indispensable de raffiner notre distinction et d'introduire des degrés de décidabilité en fonction du temps et de l'espace mémoire consommés par une machine de TURING pour résoudre un problème donné. C'est la tâche que se donne la théorie de la complexité.

Une machine de TURING est constituée d'une mémoire infinie sous forme de cases successives sur un ruban de papier, d'une tête de lecture et d'écriture ainsi que d'un mécanisme qui la positionne sur la bande mémoire et enfin d'un organe de contrôle qui lit le programme de la machine et l'exécute séquentiellement. Selon que la machine peut déplacer sa tête de lecture et d'écriture dans un seul sens ou dans les deux, et qu'à chaque transition d'états elle sélectionne son nouvel état de façon déterminée ou parmi un choix de possibilités, nous aurons affaire à une des quatre sous-classes aux propriétés bien définies.

En utilisant une machine de TURING comme modèle, on déterminera la complexité d'un problème donné en terme de temps, c'est-à-dire d'étapes successives et d'espace ou nombre de cases-mémoires. Pour certains problèmes on connaît à la fois la borne supérieure et inférieure des ressources nécessaires à leur résolution ; et plus généralement, on classe un problème implicitement en reliant sa complexité à celle d'un problème connu. On mesurera la consommation des ressources de temps et d'espace en fonction de la longueur des données en entrée. Il n'est pas besoin pour notre exposé d'aujourd'hui d'entrer plus avant dans la théorie de la complexité, il nous suffira de dire que selon que cette fonction est exponentielle ou polynomiale, on obtiendra des résultats très différents.

Ainsi une fonction exponentielle de l'ordre de 2^n est-elle rapidement inutilisable dès que n grandit.

En calcul des propositions, déterminer la validité d'une phrase est un problème parfaitement décidable : il suffit de faire une table de vérité qui prenne en compte chaque possibilité d'assignation de valeur de vérité aux différentes variables. S'il y a une seule variable, il y a deux possibilités ; s'il y a deux variables, on aura quatre assignations possibles à tester, et ainsi de suite. Si un raisonnement comporte plus de 137 variables, ce qui est parfaitement réaliste compte tenu qu'une base de données peut comporter plus d'un million de faits élémentaires, la validité du raisonnement est pratiquement invérifiable: en effet même si on pouvait tester une combinaison en un temps de 10^{-23} s, ce qui est le temps que met un photon à traverser un proton (!), il faudrait néanmoins une durée supérieure à l'âge de l'univers pour les tester toutes.

L'explosion combinatoire fait que pratiquement nous sommes limités à ne traiter que des problèmes pour lesquels la complexité est polynomiale. La progression technique aussi bien dans la rapidité des processeurs et des mémoires que dans la parallélisation massive des processeurs ne peut faire gagner que quelques ordres de grandeur.

Au-delà du critère de calculabilité, le logicien doit désormais tenir compte du critère incontournable de praticabilité. Si la thèse de CHURCH-TURING dit que tout ce qui peut être calculé peut l'être par une machine de TURING, nous suivrons H.J. LEVESQUE qui ajoute à cette thèse la restriction due à la complexité :

Tout ce qui peut être calculé peut l'être par une machine deTURING restreinte à un algorithme polynomial.

Il semble bien que le logicien soit condamné à abandonner sinon le traitement automatique, au moins les prétentions de complétude et à se limiter à des problèmes dont la complexité calculatoire n'est pas trop grande.

3 Les perspectives

Examinons en particulier le cas des bases de connaissances. On construit un programme informatique chargé de mémoriser un certain nombre de propositions, les connaissances élémentaires, et de répondre à des questions en raisonnant sur les connaissances acquises. On obtient une simulation de nos facultés cognitives et plus spécifiquement de nos processus de traitement de l'information. Ces bases de connaissances travaillent au niveau symbolique et codent le plus souvent leurs connaissances dans le langage de la logique du premier ordre. Les critères de complexité que nous venons d'évoquer imposent, aussi bien pour les modèles cognitifs du cerveau que pour les implémentations effectives, l'utilisation d'algorithmes en temps polynomial, ce qui réduit considérablement la marge de manœuvre, comme nous allons le voir, et conduit à des restrictions théoriques et à l'exploration de nouvelles voies.

Une base de connaissances formalisée en logique du premier ordre raisonne en découvrant ou en vérifiant que certaines phrases sont dans une relation d'implication ou de déductibilité à partir de l'ensemble des connaissances élémentaires et de prémisses particulières. On peut donc ramener le coeur de cette base de connaissance à un algorithme de vérification de la déductibilité entre des propositions.

Or, pour être utilisable, une telle base doit être suffisamment expressive. On peut difficilement se passer de la négation et de la disjonction: mais alors l'algorithme de vérification est déjà impraticable, et la difficulté s'accroît encore si on utilise les quantificateurs.

Dès lors plusieurs voies s'offrent à nous: la première et la plus radicale consiste à abandonner la représentation logique des connaissances au niveau symbolique et à rechercher d'autres approches, nous n'en parlerons donc pas ici; une seconde option est de conserver la représentation logique, mais d'en affaiblir la puissance par plusieurs méthodes, ce qui permet de se rapprocher de la vraisemblance psychologique dans l'élaboration d'un modèle cognitif et malgré tout de conserver un outil rigoureux qui peut résoudre la plupart des problèmes qui se posent. C'est cette seconde voie que nous allons évoquer dans la suite de notre exposé.

Une première possibilité (cf. GILLET GOCHET) consiste à restreindre la diversité des choix dans les raisonnements en associant à l'opérateur de connaissance une fonction d'attention ou de conscience qui chez Lakemeyer et Levesque accepte les situations incomplètes et incohérentes, chez Fagin et Halpern limite les données de base par un sous-ensemble qui ne laisse filtrer que les atomes dont on est conscient, ou qui chez nous tient compte de la profondeur jusqu'à laquelle un agent peut porter son attention et analyser un problème complexe.

LEVESQUE a également proposé de se restreindre à une certaine forme syntaxique qu'il appelle vivide : cela consiste essentiellement à rendre la base de connaissances complète en n'utilisant que des phrases atomiques et en établissant des conditions sur les individus. Cela a pour effet de réduire la complexité du problème de la déductibilité à celui de la recherche dans une base de données.

Bien que sous cette forme une base de connaissances manque d' expressivité, elle est déjà utilisable ; on peut augmenter l' expressivité en y adjoignant des énoncés sous forme de clauses de HORN. On peut alors exprimer, comme dans les systèmes experts, des règles de type

$$(x_1) \ldots (x_N) \{(p_1 \wedge \ldots \wedge p_K) \supset p_{K+1}\}.$$

L'expressivité obtenue ne permet pas encore d'introduire des connaissances sous la forme de disjonctions, par exemple :

Il fait 19 ou 20 degrés

ou encore :

Il fait 19 degrés ou il pleut

Alors que manifestement le deuxième exemple oblige à raisonner par cas et ainsi double la complexité, le premier lui est subsumable par un fait tel :

Il fait une petite vingtaine de degrés

Il suffit alors de remplacer la disjonction $p \vee q$ par la conjonction:

$$(p \supset r) \wedge (q \supset r),$$

où r subsume la disjonction. On augmente par cette approche l'expressivité sans toucher à la complexité.

Il existe d'autres approches qui consistent également à restreindre la puissance de la logique utilisée, mais cette fois-ci en affaiblissant la relation de déductibilité plutôt que l'expressivité du langage : l'utilisation de la logique pertinente ou de l'implication stricte suivent cette voie. Une autre possibilité consiste à utiliser des algorithmes rapides mais incomplets : une conclusion ne sera pas toujours trouvée, mais le temps de calcul sera limité dans tous les cas.

Toutes ces méthodes ne sont pas absolument générales : l'une sera plus efficace que l'autre pour un problème particulier et moins efficace pour un autre. Il nous faut développer des méthodes adaptées à chaque situation et diagnostiquer le plus précisément les problèmes à traiter. Ce n'est pas un abandon de la portée générale et théorique de la logique qui se profile, mais au contraire toute une typologie extrêmement abstraite et précise qui peut identifier avec précision les causes majeures de l'accroissement de la complexité.

4. Conclusion

Il ne faudrait pas en conclure que les problèmes de complexité et de praticabilité matérialisent des frontières infranchissables à l'intérieur desquelles la logique serait condamnée à vivoter. Au contraire, la logique est typiquement la discipline qui étudie les limites internes du raisonnement et qui, quand elle en découvre de nouvelles, enrichit son objet. L'étude approfondie de ces limites n'ambitionne pas de les effacer et de restaurer un âge d'or mythique, mais bien de construire la théorie des mécanismes qui y sont à l'œuvre. Une des pistes de recherche qui semble prometteuse est l'étude des niveaux de raisonnement.

Un problème trop complexe peut donner lieu à un méta-raisonnement sur les procédures à suivre pour le résoudre, démarche qui peut à son tour donner lieu à un méta-raisonnement. Une telle ascension doit conduire pour être efficace à une forme de convergence vers un problème de plus en plus simple qui puisse finalement être traité pratiquement.

Parallèlement, il faut bien remarquer qu'un programme incomplet ou qui manque d'expressivité peut néanmoins coder le problème à résoudre et le traiter à un niveau supérieur. Ainsi le langage de programmation PROLOG est-il basé sur un algorithme de résolution qui n'est complet que pour les clauses de HORN ; cependant il est tout à fait possible, par exemple, d'écrire dans ce langage un algorithme complet pour toutes les expressions du calcul des propositions. Le langage PROLOG ne travaille plus sur la résolution du problème proprement dit, mais sur l'exécution d'un algorithme chargé de le résoudre. De même, on peut écrire en PROLOG un programme qui simule une machine de TURING capable de traiter toute la classe des fonctions récursives, et donc résoudre tout ce qui est calculable.

Il me reste à dire que, si les répercussions mutuelles de la logique et des sciences de l'ordinateur sont nombreuses et parfois inextricables, les deux disciplines demeurent pourtant autonomes et spécifiques sinon par leur objet dans le cas présent, au moins par leur point de vue, leur méthodologie et leur finalité.

L'Outil et le Temps du Logicien: questions de François BEETS à Eric GILLET

Je voudrais poser deux questions à É. GILLET.

I. La première sera de savoir si le titre donné à sa conférence n'est pas trop modeste eu égard à ce qu'il nous y révèle? L'ordinateur n'apparaît-il pas, en effet, comme beaucoup plus qu'un simple outil? Le logicien d'avant l'ordinateur travaillait sur un univers idéal, sans aucune contrainte temporelle, et, disposant de l'éternité, pouvait utiliser les notions de calculabilité et de décidabilité. Le logicien qui s'appuie sur un outil informatique pour démontrer automatiquement certaines propositions vraies, se trouve confronté à des problèmes de complexité, calculés en termes de temps, qui rendent obsolètes les notions de calculabilité et de décidabilité. La démarche du logicien se trouve donc profondément modifiée par le recours à l'informatique, plus que par un simple outil.

II. Ma seconde question s'articule sur cette dimension temporelle que le recours à l'informatique introduit dans la problématique du logicien. Outre en effet les problèmes de complexité auxquels se heurte dorénavant le logicien, l'introduction de la dimension temporelle ne modifie-t-elle pas l'interprétation que l'on doit adopter pour certains connecteurs, et notamment pour l'implication?

Réponses d'Éric GILLET aux questions de François BEETS.

ad I. Il est vrai que le terme outil est ici une figure. Il ne s'agit pas d'un simple ustensile, mais plus généralement d'une méthodologie spécifique qui introduit ses propres contraintes tant matérielles que conceptuelles. Si le métier du logicien se trouve désormais bouleversé par cette innovation, sa démarche ne s'en trouve pas pour autant modifiée. Il n'y a pas obsolescence des concepts de la logique pré-informatique, il y a simplement de nouveaux domaines d'application qui introduisent de nouvelles problématiques et enrichissent ainsi la discipline. J'ai essayé de montrer que l'évolution des concepts et des théories au contact de l'ordinateur était le résultat d'une démarche proprement philosophique et logique au-delà du progrès purement technologique. La logique n'est pas qu'une technologie chargée de fournir des concepts et des modèles, c'est aussi une science dont l'objet est d'étudier et d'éclairer les rapports formels à l'œuvre dans toute activité rationnelle.

ad II. L'implication, signifiant la vérité d'un conditionnel quelle que soit l'interprétation des phrases liées, est un connecteur méta-linguistique qui se situe au niveau du modèle et non de la phrase. Si le modèle utilisé incorpore la notion de temps, l'implication l'incorpore implicitement via l'interprétation donnée aux phrases. Si l'on désire incorporer explicitement un rapport temporel dans la relation qui unit l'antécédent et le conséquent, on se dirige vers une tout autre notion de l'implication que celle utilisée au vingtième siècle dans le cadre de la logique mathématique. Une telle théorie est évidemment possible, mais la question se pose alors d'en justifier l'intérêt face à un modèle qui utiliserait les primitives de la logique classique et définirait ensuite des règles dérivées incorporant des contraintes temporelles.

Bibliographie sommaire

R. FAGIN & J.Y. HALPERN: "Belief, Awareness and Limited Reasoning"; in *Artificial Intelligence*, 34, 3976; (1988).

É. GILLET & P. GOCHET: "La logique de la connaissance. Le problème de l'omniscience logique"; à paraître; (1990).

D. HAREL: *Algorithmics: The Spirit of Computing*. Addison-Wesley, Don Mills, Ontario; (1986).

G. LAKEMEYER & H. J. LEVESQUE: "A Tractable Knowledge Representation Service with Full Introspection"; in *Proceedings of the 2nd Conference on Theoretical Aspects of Reasonning about Knowledge*, pp. 145-159; (1988).

H. J. LEVESQUE: Logic and the complexity of reasoning; in *Journal of Philosophical Logic*,17, pp. 355-389; (1988).

L. STOCKMEYER: "Classifying the Computational Complexity of Problems"; in *Journal of Symbolic Logic*, 52.1, pp.1-43; (1987).