



MARC PORCHERON

Jacques Pitrat, la métaconnaissance et le bootstrap de l'IA

Volume 3, n° 1-2 (2022), p. 155-166.

http://roia.centre-mersenne.org/item?id=ROIA_2022__3_1-2_155_0

© Association pour la diffusion de la recherche francophone en intelligence artificielle et les auteurs, 2022, certains droits réservés.



Cet article est diffusé sous la licence

CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.

<http://creativecommons.org/licenses/by/4.0/>



*La Revue Ouverte d'Intelligence Artificielle est membre du
Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org

Jacques Pitrat, la métaconnaissance et le bootstrap de l'IA

Marc Porcheron^a

^a EDF R&D. EDF Lab Paris-Saclay Boulevard Gaspard Monge, 91120 Palaiseau, FRANCE

E-mail : marc.porcheron@edf.fr.

RÉSUMÉ. — La *métaconnaissance* et le *bootstrap des connaissances* sont deux notions au cœur de la conception originale de l'IA de Jacques Pitrat, qu'il a élaborées et étudiées tout au long de sa carrière. Ce texte en présente les principes ainsi qu'une illustration tirée du travail de thèse que j'ai eu la chance de réaliser sous sa direction. Il aborde également le rôle clé joué dans l'esprit de Jacques Pitrat par ces deux concepts dans l'optique d'une possible *singularité* qui verrait les capacités de l'Intelligence Artificielle dépasser celles de l'Intelligence Humaine.

MOTS-CLÉS. — Métaconnaissance, *bootstrap*, singularité.

1. INTRODUCTION

S'il est un concept qui aura passionné Jacques Pitrat durant toute sa carrière de chercheur et d'enseignant en IA, c'est bien celui de *métaconnaissance*. Pour lui, la capacité à la *réflexivité*, c'est à dire à l'analyse de ses propres connaissances et de ses propres processus cognitifs dans le but de les enrichir et d'en améliorer les performances, est une faculté clé, sinon caractéristique, de toute forme d'intelligence, qu'elle soit humaine ou artificielle.

Cette faculté semble en effet ouvrir la voie à un processus d'« auto-incrémentation » des systèmes cognitifs, dès lors que ces métaconnaissances peuvent s'appliquer à elles-mêmes. Ainsi, un système artificiel disposant de connaissances pour découvrir de nouvelles connaissances pourrait, en les appliquant à elles-mêmes, découvrir de nouvelles connaissances pour découvrir de nouvelles connaissances, augmentant ainsi ses capacités dans un processus *a priori* sans limites, ou du moins dont les limites seraient, affirmait Jacques Pitrat, comparables à celles rencontrées par un cerveau humain. La métaconnaissance serait ainsi la clé d'une possible *singularité* qui verrait les capacités de l'Intelligence Artificielle égaler, voire dépasser, celles de l'Intelligence Humaine.

Dès lors que l'on s'intéresse à ce concept de « connaissances portant sur des connaissances et s'appliquant à elles-mêmes » on est immédiatement confronté au

problème de l'*amorçage* : comment initier un tel processus dès lors qu'au début nous ne disposons pas de ces métaconnaissances ? Jacques Pitrat a consacré une partie importante de ses recherches à ce problème, et a démontré qu'il pouvait être résolu en appliquant des techniques de *bootstrapping* incrémentales, inspirées de celles classiquement mises en œuvre pour écrire le compilateur d'un langage dans le langage qu'il est destiné à compiler.

Cet article propose une présentation de ces différents concepts, tels que Jacques Pitrat les a étudiés non seulement d'un point de vue théorique, mais également d'un point de vue expérimental en les mettant concrètement en œuvre dans les nombreux systèmes qu'il a lui-même développés tout au long de sa carrière, ou dans ceux développés sous sa direction par ses doctorants dont j'ai eu la chance de faire partie à la fin des années quatre-vingt.

La section 2 traite de la *métaconnaissance* et du rôle clé qu'elle est censée jouer dans l'esprit de Jacques Pitrat pour permettre à l'IA de dépasser ses limites ; y sont également présentés les principes de formalisation des connaissances et des métaconnaissances dans les langages à base de règles, avec l'idée majeure que ce formalisme peut servir aussi bien à représenter des connaissances sur un domaine d'expertise donné que des métaconnaissances portant sur ces connaissances. La section 3 traite du problème du *bootstrap*, quelques résultats obtenus dans le cadre de la thèse que j'ai réalisée sous la direction de Jacques Pitrat servant d'illustration. La section 4 aborde la question de la *singularité* sous l'angle du rôle possible de la métaconnaissance que Jacques Pitrat envisageait dans ce contexte. Enfin, la section 5 s'interroge en guise de conclusion sur de possibles liens fructueux entre la notion de *bootstrap* de l'IA dans sa version « symbolique », telle que Jacques Pitrat l'a investiguée, et la notion de *bootstrap* de l'IA dans sa version « connexionniste/neuronale » qui a connu des avancées remarquables ces dernières années.

Sur la forme, il m'a semblé approprié dans ce numéro spécial d'hommage à Jacques Pitrat de présenter ces concepts comme il le faisait lui-même, et c'est donc le plus souvent en s'appuyant sur des citations tirées de ses publications qu'ils sont introduits dans la suite.

2. MÉTACONNAISSANCE, FUTUR DE L'IA

« Des systèmes doués d'une intelligence supérieure doivent être capables de connaître le domaine dans lequel ils travaillent, mais aussi d'examiner ce qu'ils doivent faire, ce qu'ils peuvent faire, ce qu'ils savent, ce qu'ils sont en train de faire. Ils opèrent à un niveau "méta", au-dessus du niveau du problème à traiter ; la connaissance devient pour eux objet d'étude. La **métaconnaissance** est précisément l'outil qui permet de travailler sur la connaissance et de réaliser de tels systèmes supérieurement intelligents. » Jacques Pitrat, in *Métaconnaissance, futur de l'intelligence artificielle* [11].

2.1. UNE CONVICTION TRÈS FORTE ET TRÈS ANCIENNE

En 1979, dans un texte écrit en réponse au célèbre livre de Hubert Dreyfus « Intelligence Artificielle. Mythes et limites » [2], Jacques Pitrat distinguait deux auteurs dans le livre en question.

Le premier, avec lequel il était bien sûr en complet désaccord, voyait des limites aux possibilités des ordinateurs ; à ce premier Hubert Dreyfus, Jacques Pitrat répondait qu'on ne pouvait démontrer que ces limites « *ne seraient pas aussi des limites aux possibilités de l'esprit humain* ».

Le second affirmait que les approches investiguées pour dépasser les limites de l'IA à l'époque étaient vouées à l'échec. Et avec ce second Hubert Dreyfus, Jacques Pitrat se trouvait parfaitement d'accord.

En particulier, Jacques Pitrat exprimait ses doutes quant aux possibilités de succès de la voie que l'IA allait emprunter très majoritairement dans les années suivantes dans le cadre du développement des *systèmes experts* : celle consistant à essayer d'acquérir auprès des experts humains les connaissances nécessaires à la résolution de problèmes complexes :

« Il n'est pas facile de trouver les connaissances que nous utilisons : elles sont souvent inconscientes. De plus, nous les stockons en mémoire sous une forme telle que nous ne pouvons les donner spontanément : elles sont déclenchées par les problèmes posés, nous ne pouvons les énumérer. » [10].

Puis il écrivait, donnant ici la solution qui selon lui allait permettre de dépasser ces limites et qu'il allait explorer tout au long de sa carrière :

« Comme nous avons de la peine à extraire les connaissances d'un expert humain, il faudra que nous en venions à la réalisation de programmes capables de les découvrir automatiquement, donc capables d'apprendre. Pour cela, ils devront disposer de connaissances pour trouver des connaissances. L'autre difficulté est de réaliser des programmes capables d'utiliser efficacement de grandes quantités de connaissances. Il n'est pas possible pratiquement d'incorporer les connaissances au programme : celui-ci serait gigantesque, donc pratiquement impossible à mettre au point et à modifier. Il faut pouvoir les donner « en vrac », de façon à ce que nous puissions en ajouter, les enlever ou les modifier facilement. Mais comment réaliser un programme qui utilise avec efficacité des connaissances données sous cette forme ? Je pense que la solution est de prendre pour cela un programme de même nature qui dispose de connaissances pour utiliser efficacement des connaissances. **Ce sont des métaconnaissances.** » [10].

2.2. LES RÈGLES D'INFÉRENCES, MODÈLE DE LA CONNAISSANCE HUMAINE ... ET COMPUTATIONNELLE

Pour Jacques Pitrat, les systèmes formels sont la forme mathématisée par excellence de la connaissance humaine. Les axiomes décrivent nos connaissances sur le monde, qu'elles soient purement *factuelles* comme l'assertion : « Bertrand Russel est un logicien » ou *déductives* comme l'assertion : « Tous les logiciens sont bizarres », et les *règles d'inférences* permettent de déduire de nouvelles connaissances à partir des axiomes.

Le *calcul des prédicats du premier ordre* permet de formaliser le processus déductif à l'œuvre sur la base de ce type de connaissance purement *déclarative*. Ainsi, des axiomes :

$$\text{Logicien}(\text{Russel})$$

$$\forall x, \text{Logicien}(x) \supset \text{Bizarre}(x),$$

les règles de *substitution* et de *détachement* (*modus ponens*) permettent de déduire la nouvelle connaissance factuelle « Russel est bizarre » :

$$\text{Bizarre}(\text{Russel}).$$

L'automatisation de ce processus déductif a été à la base des travaux fondateurs de l'IA dite « symbolique » dans les années cinquante et soixante, appliqués à la démonstration automatique de théorèmes (cf. par exemple le « Logic Theorist » de A. Newell et H. Simon considéré comme le premier programme d'IA [8], ou la thèse d'État de Jacques Pitrat lui-même, soutenue en 1966 [9]).

À partir du début des années soixante-dix, elle a constitué le principe de base des *systèmes experts* (cf. par exemple [3] pour une revue des premiers systèmes de ce type, dont les célèbres DENDRAL et MYCIN).

Dans ces systèmes, l'assertion factuelle « *Logicien(Russel)* » trouve sa place dans la *base de faits*, l'assertion déductive « $\forall x, \text{Logicien}(x) \supset \text{Bizarre}(x)$ » trouve sa place dans la *base de règles*, classiquement sous la forme :

$$\text{SI } \text{Logicien}(x) \text{ ALORS } \text{Bizarre}(x),$$

tandis que le *moteur d'inférences* est en charge du *modus ponens*.

Cette approche a donné lieu à de très nombreuses applications dans les années quatre-vingt, avec des succès indéniables dans des domaines variés (industrie, finance, médecine ...), avant que des limites finissent par en circonscrire l'expansion.

Et comme en témoignent les citations de la section précédente, Jacques Pitrat avait parfaitement pressenti que parmi ces limites se trouvait la difficulté à encoder dans une base de connaissances la totalité du savoir d'un expert humain sur un domaine et, à supposer qu'on y parvienne, à utiliser efficacement les grandes quantités de connaissances relatives à une telle expertise.

Pour lui, il était vain d'espérer y parvenir par un patient travail d'extraction/élicitation des connaissances expertes, puis de mise au point d'un programme capable de les

utiliser efficacement. La solution se trouve au contraire dans la fourniture au programme de *métaconnaissances* lui permettant d'une part d'apprendre automatiquement de nouvelles connaissances, et d'autre part d'utiliser efficacement celles-ci. De plus, ces métaconnaissances ne peuvent être fournies au programme autrement qu'« en vrac », comme il le dit ci-dessus, c'est à dire sans mode d'emploi procédural mais de manière *déclarative*, dans le même formalisme que les connaissances de base sur le domaine d'expertise considéré.

Et le formalisme du calcul des prédicats s'y prête parfaitement : après tout, qu'est-ce qu'une règle, sinon une connaissance comme une autre, que l'on peut décrire et sur laquelle on peut raisonner ?

Le second type de métaconnaissances évoqué ci-dessus, les connaissances pour utiliser efficacement des connaissances, trouve en effet une formalisation toute naturelle dans ce modèle. Les deux *méta-règles* ci-dessous illustrent cette idée :

- « Si $R1$ et $R2$ sont des règles et si le nombre de prémisses de $R1$ est inférieur à celui de $R2$, alors $R1$ est plus facile à évaluer que $R2$. »
- « Si $R1$ et $R2$ sont des règles et si le nombre de conclusions de $R1$ est inférieur à celui de $R2$, alors $R1$ est moins intéressante à évaluer que $R2$. »

Il suffit de décrire dans la base de faits les caractéristiques des règles : nombre de prémisses, de conclusions, etc., pour que le moteur d'inférences déduise, à partir de ces *méta-faits* et des méta-règles ci-dessus, de nouvelles connaissances directement exploitables pour améliorer l'exploitation de la base de connaissances.

Identifier et formaliser des connaissances pour apprendre de nouvelles connaissances représente un défi autrement plus difficile à relever, au cœur de la vision de l'IA de Jacques Pitrat. Je me souviens de cet exemple qu'il prenait dans son cours de DEA à l'Université Pierre et Marie Curie que j'ai eu la chance de suivre au début des années quatre-vingt, tiré des programmes AM et Eurisko de Douglas Lenat [6, 7] :

- « Si un concept repose sur une notion d'ensemble, alors étudier les concepts correspondant aux cas extrêmes, c'est-à-dire ceux où cet ensemble comporte un très petit nombre d'éléments, ou au contraire un très grand nombre d'éléments. »

Si on considère le concept : « Ensemble D des diviseurs d'un entier », et si on s'intéresse au cas où $|D|$ est très grand, on ne découvre a priori rien d'intéressant ; par contre, si on s'intéresse au cas où $|D|$ est très petit . . . on a découvert le concept de nombre premier !

Il existe donc bien des *heuristiques* de la découverte, i.e. des connaissances qui permettent d'apprendre, de découvrir, et de synthétiser de nouvelles connaissances. C'est ce sujet qui a toujours passionné Jacques Pitrat, qui y voyait la clé d'une possible « auto-incrémentation » de l'IA conduisant à la fameuse « singularité » dont nous reparlerons plus loin.

3. RÉFLEXIVITÉ/AUTORÉFÉRENCE, BOOTSTRAP/AMORÇAGE

« Le but d'un amorçage est de mettre en œuvre un objet à l'aide de lui-même. Cette définition apparaît à première vue paradoxale, car comment se servir d'un objet qui n'existe pas encore. Pour comprendre l'amorçage, nous devons faire intervenir le facteur temps. Il y a en réalité une succession d'objets de plus en plus perfectionnés et chacun participe à l'élaboration de son successeur. Vu sous cet angle, le paradoxe disparaît. Nous utilisons très souvent cette méthode pour réaliser des tâches complexes. L'amorçage est utile dans toutes les situations où nous nous disons : "Je ferais plus facilement ce que je suis en train de faire si je l'avais déjà fait". »

Jacques Pitrat, in *Des métaconnaissances pour des systèmes intelligents* [12].

Le *bootstrap*, ou *amorçage*, est une technique bien connue des informaticiens pour écrire un compilateur dans le même langage que celui qu'il est destiné à compiler.

Soit un langage (L) dont on veut écrire un compilateur en (L); c'est a priori impossible, sauf à disposer du pouvoir du légendaire Baron de Munchausen qui pouvait se soulever lui-même en tirant sur les sangles de ses bottes (c'est l'origine du terme « bootstrap ») ...

Pourtant, procéder par étapes permet de résoudre ce paradoxe apparent, et Jacques Pitrat a découvert que cette technique, habituellement réservée aux langages classiques procéduraux, pouvait tout aussi bien s'appliquer aux langages déclaratifs à base de règles.

Les *T-Diagrams* [1] sont un formalisme graphique bien adapté à la représentation d'un tel processus. Sur la figure 3.1, le diagramme de gauche représente le cas standard où un compilateur écrit dans un langage (C) compile le langage (L) en un langage (M) exécutable sur la machine (éventuellement après avoir été lui-même compilé); le diagramme de droite illustre la situation a priori paradoxale où un compilateur écrit en (L) compile le langage (L).



FIGURE 3.1. T-Diagrams. À gauche, le cas standard où un compilateur écrit en (C) compile le langage (L) en (M); à droite, on veut écrire en (L) un compilateur du langage (L).

Ce paradoxe apparent est résolu par les étapes suivantes :

- (1) **Amorçage** : écrire en (M) un compilateur $C0$ pour un sous-langage (L0) de (L); par exemple (L0) ne contient que des boucles *for* et pas de boucles *while* (Figure 3.2).



FIGURE 3.2. Amorçage : on écrit en (M) un compilateur $C0$ de (L0).

- (2) **Extension**
 (a) Étendre (L0) en (L1) \subset (L), et écrire le compilateur $C1$ de (L1) en (L0) (Figure 3.3);



FIGURE 3.3. Extension (a). On écrit en (L0) un compilateur $C1$ de (L1)

- (b) Appliquer $C0$ à $C1$. On obtient un compilateur $C2$ de (L1) (Figure 3.4).

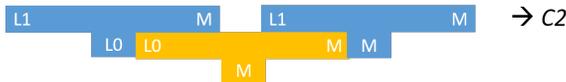


FIGURE 3.4. Extension (b). On applique $C0$ à $C1$. On obtient un compilateur $C2$ de (L1).

- (3) **Répéter** les opérations (2a) et (2b) jusqu'à obtenir le compilateur final de (L) écrit en (L) (Figure 3.5).



FIGURE 3.5. Le compilateur final de (L) écrit en (L).

Jacques Pitrat a montré que ces principes pouvaient s'appliquer au *bootstrap* de la métaconnaissance, en utilisant un « moteur de démarrage » pour la phase d'amorçage. Comme dans le cas procédural, le langage de règles initial ne comporte pas toutes les possibilités du langage de règles final, par exemple le type de prémisses de règles exploitables est limité, et le moteur d'amorçage exploite de l'information procédurale, par exemple en exécutant les règles dans l'ordre où elles sont déclarées, alors que le

moteur final décidera lui-même de l'ordre d'exécution le plus efficace sur la base de la méta-expertise de compilation.

Ce sont ces idées que j'ai essayé de mettre en pratique sous sa direction dans ma thèse, en construisant une méta-expertise de compilation d'un langage de règles fondé sur le calcul des prédicats [17]. Cette méta-expertise comportait 500 méta-règles, dont 100 utilisées pour la génération du langage initial d'amorçage et 400 pour l'analyse et la compilation du langage final. Ces dernières réalisaient les opérations suivantes sur une base de règles quelconque :

- Analyse des interactions entre règles et construction d'un « graphe de propagation » modélisant la relation d'influence entre règles ;
- Décomposition du graphe en composantes connexes et ordonnancement de celles-ci par la fonction de rang ;
- Décision d'heuristiques d'exécution efficaces en fonction de l'analyse (degré de mémorisation d'information utile, génération de code exécutant les parties « procédurales » éventuellement détectées dans la base de connaissances ...).

Voici un exemple de règle du langage, exprimant la transitivité de la relation « être amis de », modulo une condition sur l'âge :

```
[R1 isa rule
lhs
  [?y belongs [?x friends]]
  [?z belongs [?y friends]]
  [?u = [?z age]]
  [?u > 20]
rhs
  [?x friends add ?z]
]
```

Et voici un exemple de méta-règle du langage, exprimant dans le même formalisme une relation d'influence entre deux règles sur la base de leur partie prémisses/conclusions respectives :

```
[MR1 isa rule
lhs
  [?c belongs [?r1 rhs]]
  [?p belongs [?r2 lhs]]
  [[?c predicat]=[?p predicat]]
  [[?c action]= "add"]
rhs
  [?r1 influence+ add ?r2]
]
```

MR1 est ainsi capable de détecter une influence de R1 sur elle-même, consistant à ajouter dans la base de faits des assertions susceptibles d'instancier ses prémisses, et donc un risque de bouclage lors de son interprétation par le moteur d'inférences.

J'ai tout d'abord testé cette méta-expertise de compilation sur les bases de connaissances d'un prototype de système expert développé par la Compagnie Générale d'Électricité où je faisais ma thèse dans le cadre d'une collaboration avec le Laforia. Les performances de saturation de ces bases (i.e. d'exécution de toutes les inférences possibles à partir de la base de faits initiale) obtenues après leur compilation par cette méta-expertise étaient bien meilleures que celles obtenues par un moteur d'inférences classique.

Je me revois présenter ces résultats à Jacques Pitrat et je l'entends encore me dire, ce à quoi je m'attendais un peu ... :

« *C'est très bien, mais maintenant il faut appliquer la méta-expertise à elle-même.* »

J'ai donc appliqué, avec le moteur d'inférences final, la méta-expertise de compilation à la quarantaine de méta-règles chargées de la compilation des parties conditions des règles du langage final, et comparé les performances avec celles obtenues d'une part par un moteur d'inférences classique, et d'autre part par le moteur d'inférences initial de *bootstrap* exploitant la version de cette méta-expertise écrite dans le langage initial et tirant parti d'informations procédurales de contrôle (évaluation des règles dans l'ordre de leur déclaration dans la base, évaluation des conditions d'une règle dans l'ordre de leur déclaration dans la règle. . .).

Là aussi les résultats ont été très probants. D'une part, l'exécution par le moteur final de la méta-expertise était nettement plus efficace que par le moteur d'inférences classique, confirmant les résultats obtenus lors de la première expérimentation, mais les temps d'exécution par le moteur final de la méta-expertise après compilation par elle-même étaient même un peu meilleurs que ceux obtenus en appliquant le moteur de *bootstrap* à la même expertise écrite dans le langage initial. Le système s'était donc bien « auto-amélioré » en générant une méta-expertise plus efficace et plus déclarative que celle qu'on lui avait fournie au départ, dans la phase d'amorçage.

4. LA MÉTACONNAISSANCE, CLÉ DE LA SINGULARITÉ

« Supposons qu'existe une machine surpassant en intelligence tout ce dont est capable un homme, aussi brillant soit-il. La conception de telles machines faisant partie des activités intellectuelles, cette machine pourrait à son tour créer des machines meilleures qu'elle-même ; cela aurait sans nul doute pour effet une réaction en chaîne de développement de l'intelligence, pendant que l'intelligence humaine resterait presque sur place. Il en résulte que la machine ultra intelligente sera la dernière invention que l'homme aura besoin de faire, à condition que ladite machine soit assez docile pour constamment lui obéir. »

La citation ci-dessus date de 1965 et est le fait d'Irving John Good, membre de l'équipe d'Alan Turing sur le projet Enigma de 1941 à 1942. Elle est considérée comme une des

toutes premières expressions de la notion de *singularité*, c'est à dire d'un point où l'IA deviendrait « autonome » capable de s'auto-améliorer dans un processus permanent, laissant l'intelligence humaine stagner loin derrière elle.

Il ne rentre pas dans le cadre de cet article de discuter des possibilités et des implications d'un tel événement, qui suscitent de nombreux débats (cf. par exemple [4, 20], ou l'opinion de Jacques Pitrat lui-même sur la question dans [16]).

Ce qui est certain, c'est que Jacques Pitrat croyait à la possibilité d'une telle singularité, précisément fondée sur l'utilisation de métaconnaissances et sur le *bootstrap*/amorçage de l'IA, comme en témoignent les deux citations ci-dessous qui font écho trente ans plus tard à celle de d'Irving John Good :

« J'avais suggéré que l'IA pouvait nous aider à mettre en œuvre l'IA. Ce n'est possible que si nous rendons l'IA réflexive, capable de s'appliquer à elle-même. Cela est le cas à partir du moment où l'on introduit les métaconnaissances. Nous trouverons de nouvelles connaissances pour découvrir des connaissances à l'aide des connaissances pour découvrir des connaissances déjà existantes. Nous pouvons ainsi amorcer l'IA. »

Jacques Pitrat, in *Des métaconnaissances pour des systèmes intelligents* [12].

« Un avenir à très long terme [...] Une critique classique, mais mal fondée, contre l'IA est la constatation que l'ordinateur ne fait jamais que ce qu'on lui dit de faire. Tout le monde en est convaincu ! Dans l'approche classique de l'IA, on veut justement lui dire de se comporter intelligemment. Le problème est que nous ne savons pas encore bien comment le lui dire. Avec l'approche basée sur la métaconnaissance, nous voulons lui dire de se comporter de plus en plus intelligemment. Nous espérons amasser assez de métaconnaissances pour arriver à des systèmes capables de s'améliorer. Cette progression sera lente au départ. Mais, dans ce schéma, tout progrès amène une accélération des progrès ultérieurs. Notre but est que l'IA assure elle-même la plus grande partie de la construction de son état final. »

Jacques Pitrat, in *Des métaconnaissances pour des systèmes intelligents* [12].

5. CONCLUSION

La métaconnaissance, la réflexivité, et le *bootstrap* des connaissances sont des concepts fondamentaux pour l'IA que Jacques Pitrat a inlassablement investigués tout au long de sa carrière. Au-delà de leur rôle, pour lui central, dans le développement de l'IA, Jacques Pitrat voyait même dans ces concepts une base possible à la définition d'une forme de « conscience artificielle » [15].

Il ne s'est pas contenté de les étudier de manière théorique, ni au travers des nombreuses thèses qu'il a dirigées, mais il les a mis lui-même en pratique dans les systèmes qu'il a développés tout au long de sa vie : MACISTE [13], dont le solveur général de problèmes, MALICE [14], est inspiré des travaux de Thèse d'État de Jean-Louis Laurière [5], et CAIA, le « Chercheur Artificielle en Intelligence Artificielle » [15].

Ces travaux démontrent la pertinence de ces concepts dans différents domaines, au premier chef celui de l'amélioration des performances des systèmes d'inférences par leur auto-analyse. Mais Jacques Pitrat insistait toujours sur l'extrême difficulté à découvrir l'ensemble « minimal » de métaconnaissances d'apprentissage qui permettrait d'initier le *bootstrap* et de déclencher cette auto-construction de l'IA qu'il voyait comme la clé d'une potentielle singularité.

Bien que reposant sur des principes différents de ceux sur lesquels Jacques Pitrat a concentré ses efforts, l'IA connexionniste/neuronale est aussi un champ d'application de ses idées : en 2017, AlphaGo Zero est ainsi devenu le meilleur joueur de Go au monde en ne « connaissant » au départ que les règles du jeu, et en jouant uniquement contre lui-même [19]; AlphaZero, sa généralisation reposant sur les mêmes principes, est devenu depuis le meilleur joueur d'échecs et de shogi ...

À l'heure où l'hybridation de méthodes issues de l'IA symbolique et de l'IA connexionniste/neuronale fait l'objet de nombreux travaux (cf. [18] pour une revue des recherches en cours sur le sujet), l'étude d'une telle hybridation spécifiquement dédiée au problème du *bootstrap* de l'IA pourrait fonder un programme de recherche dans la continuité de la contribution majeure de Jacques Pitrat à l'IA et à son histoire.

BIBLIOGRAPHIE

- [1] H. BRATMAN, « An alternate form of the uncol diagram », *Communications of the ACM* **4** (1961), n° 3.
- [2] H. L. DREYFUS, *Intelligence Artificielle. Mythes et limites*, Flammarion, 1979.
- [3] E. A. FEIGENBAUM, « The art of artificial intelligence: Themes and case studies of knowledge engineering », in *IJCAI*, 1977.
- [4] J.-G. GANASCIA, *Le Mythe de la Singularité. Faut-il craindre l'intelligence artificielle ?*, Seuil, 2017.
- [5] J.-L. LAURIÈRE, « Un langage et un programme pour énoncer et résoudre des problèmes combinatoires », Thèse d'état, Université Paris 6, 1976.
- [6] D. B. LENAT, « AM: An artificial intelligence approach to discovery in mathematics as heuristic search », *SAIL AIM-286, Artificial Intelligence Laboratory, Stanford University* (1976).
- [7] ———, « EURISKO: a program that learns new heuristics and domain concepts », *Artificial Intelligence* **21** (1983), n° 1-2, p. 61-98.
- [8] A. NEWELL & H. A. SIMON, « The logic theory machine—A complex information processing system », *IRE Transactions on information theory* **2** (1956), n° 3, p. 61-79.
- [9] J. PITRAT, « Réalisation de programmes de démonstration de théorèmes utilisant des méthodes heuristiques », Thèse d'état, Faculté des Sciences de l'Université de Paris, 1966.
- [10] ———, « Quelques remarques sur *Intelligence Artificielle. Mythes et limites* », in *Intelligence Artificielle – Mythes et Limites*, Flammarion, 1979.
- [11] ———, *Métaconnaissance : futur de l'intelligence artificielle*, Hermes, 1990.
- [12] ———, « Des métaconnaissances pour des systèmes intelligents », *Quaderni* **25** (1995), p. 29-42.
- [13] ———, « Implementation of a reflective system », *Future Generation Computer Systems* **12** (1996), n° 2-3, p. 235-242.

- [14] ———, « MALICE, notre collègue », in *Colloque Métaconnaissance de Berder*, LIP6, 19-21 Septembre 2001, p. 4-19.
- [15] ———, *Artificial Beings: The conscience of a conscious machine*, Wiley-ISTE, London, 2009.
- [16] ———, « My view on Artificial Intelligence », 2019,
<http://bootstrappingartificialintelligence.fr/WordPress3/>.
- [17] M. PORCHERON, « Utilisation de méta-connaissances pour la compilation des règles de production », Thèse, Université Paris 6, 1990.
- [18] K. SAKER, L. ZHOU, A. EBERHART & P. HITZLER, « Neuro-Symbolic Artificial Intelligence Current Trends », <https://arxiv.org/abs/2105.05330>, 2021.
- [19] D. SILVER, J. SCHRITTWIESER & K. SIMONYAN, « Mastering the game of Go without human knowledge », *Nature* **550** (2017), p. 354-359.
- [20] T. WALSH, « The singularity may never be near », *AI Magazine* **38** (2017), n° 3, p. 58-62.

ABSTRACT. — *Meta-knowledge* and the *bootstrap of knowledge* are two concepts at the core of Jacques Pitrat's original conception of AI, which he developed and studied throughout his career. This text presents their principles as well as some illustration drawn from the PhD that I had the chance to carry out under his direction. It also discusses the key role played in Jacques Pitrat's mind by these two concepts in the perspective of a possible *singularity* which would see the capacities of Artificial Intelligence exceed those of Humain Intelligence.

KEYWORDS. — Metaknowledge, bootstrap, singularity.

Manuscrit reçu le 30 mars 2021, révisé le 11 août 2021, accepté le 15 septembre 2021.