

NNTP and Prolog

Shinichiro MADO*¹

*¹ Utsunomiya University

In the case of knowledge representation of agents modeled with automata, the most common style of knowledge representation is the NNTP. In this paper I will present that the programming language Prolog is very useful for the representation of NNTP. "NNTP" means "Non-Neumann Type Program". Generally speaking, a NNTP is a set of rules which can be described in the style of IF-THEN. However, if we use Prolog, we can describe it with structures and in more sophisticated styles. This is important for the representation of intelligence. In order to show this, I will discuss an extension of the model which was made by Axelrod. This extension of the model of Axelrod needs an extension of the method of GA. Especially the extension of the notion of "chromosome," that represents the gene, is necessary. In the model adopted by Axelrod, the "chromosome" is represented by a 70bit binary sequence of 0s and 1s. But this extension of the model needs another method of representation, because such a binary sequence can not describe any more sophisticated knowledge about the social environment. And as such, this extension of knowledge makes GA useless. In consequence, we must adopt another method to describe the evolution of society instead of GA. This new method to describe the evolution of society is called "ISEM". Indeed we also need a powerful programming language to describe knowledge of the IF-THEN rules, because we must describe the complex IF-THEN rules instead of binary sequences. The language Prolog allows us to cope with such complex knowledge. Additions, subtractions and any other developments of knowledge can be represented by the language Prolog. Not until we use this sophisticated language of artificial intelligence can we describe the evolution of society.

1. Introduction

Le langage informatique nommé « Prolog » est très utile pour décrire l'intelligence humaine.

Je voudrais discuter sur l'extention du modèle utilisé par Axelrod. Cette extension du modèle d'Axelrod a besoin de l'extension du GA. Donc nous avons besoin de l'autre méthode pour décrire l'évolution sociale à la place du GA. C'est la méthode nommé « ISEM ». De plus nous avons besoin du Prolog pour décrire l'ISEM. Le Prolog est nécessaire pour construire des modèles basés sur la méthode nommée « ISEM ».

Les jeux évolutifs d'Axelrod combinent le jeu du dilemme du prisonnier et l'évolution des stratégies avec l'algorithme génétique. Axelrod a construit ce modèle et fait cette simulation. Il a obtenu le résultat suivant : si les interactions durent assez longtemps, des coopérations sur les dépendances mutuelles peuvent naître et durer dans un monde d'égoïstes. C'est un résultat observable dans la politique internationale.

2. Les jeux évolutifs d'Axelrod

Le tableau ci-dessous est la matrice des gains dans un jeu du dilemme du prisonnier. Les valeurs numériques sont les gains des deux joueurs. Dans ce cas, il y a deux choix, le premier choix est la coopération notée sous la forme « C » et le second choix est de faire cavalier seul noté sous la forme « D ».

		Joueur de la colonne	
		Coopérer « C »	Faire cavalier seul « D »
Joueur de la ligne	Coopérer « C »	R=3, R=3 Récompense pour coopération mutuelle	S=0, T=5 Salaire de la dupe, et tentation de l'égoïste
	Faire cavalier seul « D »	T=5, S=0 Tentation de l'égoïste, et salaire de la dupe	P=1, P=1 Punition de l'égoïste

Figure 1. Le dilemme du prisonnier

Dans le tableau ci-dessus, les gains du joueur de la ligne sont donnés en premier. Cela signifie que les joueurs dans ce jeu ne peuvent pas changer leur environnement par eux-même. Il faut que cet environnement soit un environnement social. Mais comme cet environnement est commun à l'environnement naturel, les joueurs ne peuvent pas changer leur environnement. Alors, nous pouvons dire que cet environnement exprimé dans le tableau précédent est une représentation de l'environnement naturel.

Dans le jeu précédent, il y a les 4 cas possibles suivants.

$$\{CC, CD, DC, DD\}$$

Adresse : M. Shinichiro MADO, Maître Assistant d'Éconoinformatique, Faculté des Études Internationales, Université d'Utsunomiya, Mine-machi 350, Utsunomiya, Tochigi, 321-8505, Japon,
Mél. : mado@utsunomiya-u.ac.jp

Nous pouvons écrire les règles de l'attitude dans le tableau ci-dessous, codé en langage informatique de type « si-alors ». Quand chacun joueur suit les règles.

$$\left\{ \begin{array}{llll} Si & CC & Alors & C \\ Si & CD & Alors & D \\ Si & DC & Alors & C \\ Si & DD & Alors & D \end{array} \right.$$

Maintenant, si nous mettons la colonne, la plus à droite de ce tableau, donné ci-dessus, à part, nous pouvons alors décrire ces règles sous la formule linéaire suivante :

$$CDCD$$

Cette rangée contient 4 caractères, parce que la partie à gauche du tableau codé en langage informatique de type « si-alors » est commune à tous les joueurs.

Dans le cas de la simulation d'Axelrod, supposons que chacun joueur est un pays, alors chaque pays possède un ensemble de règles présentées ci-dessous.

$$64 \text{règles} \left\{ \begin{array}{llllll} Si & CC & CC & CC & Alors & C \\ Si & CC & CC & CD & Alors & D \\ Si & CC & CC & DC & Alors & C \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Si & DD & DD & DD & Alors & D \end{array} \right.$$

Dans le cas, où chaque pays a un ensemble de situations codées sous la forme ci-dessous, nous pouvons exprimer « C » et « D » par « 0 » et « 1 ».

$$\{C, C, C, C, D, C\} \text{ ou } \{0, 0, 0, 0, 1, 0\}$$

Ceci est l'enregistrement de choix de deux joueurs qui ont fait les trois derniers jeux. Le premier groupe de deux caractères signifie l'avant avant dernier jeu. Le premier des deux caractères dans un jeu signifie le choix qui est pris par ce joueur, et le second des deux caractères dans un jeu signifie le choix qui est pris par l'autre joueur. Ensuite le troisième et le quatrième caractère signifient le jeu avant dernier. Puis le cinquième et le sixième caractère signifient le dernier jeu.

Donc nous pouvons écrire l'ensemble des situations avec 6 caractères. Si nous prenons la colonne la plus à droite, nous pouvons écrire cet ensemble de règles sous la forme ci-dessous.

$$CDC \dots D \text{ ou } 010 \dots 1$$

C'est une ligne de 64 caractères, car la partie à gauche du tableau est commune à tous les joueurs. Donc nous pouvons écrire l'ensemble des règles de l'attitude et des états de l'enregistrement d'un joueur avec 70 caractères. Cet ensemble de 70 caractères est un NNTP.

Axelrod appliquait l'algorithme génétique à ce type de modèle en jeu.

(1) D'abord, Axelrod choisit 20 individus ayant des stratégies aléatoires. Chacune des stratégies est exprimée dans une ligne de 70 caractères.

(2) Ensuite, Axelrod fait jouer 20 individus beaucoup de fois au dilemme du prisonnier. Puis, il évalue les individus par leurs propre moyenne de résultats au jeu.

(3) Puis, il fait l'accouplement suivant les gains.

Axelrod calcule la moyenne : μ et l'écart-type : σ de profit et classe les individus en trois groupes : g_s, g_m, g_i . Le groupe moyen : g_m est dans l'étendue : $|X - \mu| \leq \sigma$. Et les individus dans ce groupe peuvent se croiser une fois. Le groupe supérieur : g_s est dans l'étendue : $X - \mu > \sigma$. Et les individus dans ce groupe peuvent se croiser deux fois. Ensuite le groupe inférieur : g_i est dans l'étendue : $X - \mu < -\sigma$. Les individus dans ce groupe ne peuvent plus se croiser.

(4) Puis, il fait un croisement génétique et une mutation.

Axelrod a croisé les individus avec la mutation à une proportion de un pour dix. Quand il croise les individus, il choisit une place dans la séquence du code génétique à la probabilité de se croiser. Et il coupe les codes génétiques des parents en deux à la place. Ensuite les côtés différents des codes des parents sont copiés au nouveau bébé. Dans le cas de la mutation, il fait des erreurs de copie des codes à la probabilité de la mutation.

(5) Après, il alterne des générations.

Après que tous les individus de nouvelle génération sont préparés, il alterne des générations.

(6) Enfin, il répète les processus de (2) à (5) plus de 50 fois.

Ce processus, qui est répété de (1) à (6) beaucoup, est l'algorithme génétique, qui est appliqué par Axelrod à sa simulation des jeux évolutifs.

3. Les qualités et les défauts de la méthode d'Axelrod

3.1 Qualité 1: Axelrod utilisait le modèle de type automate.

Axelrod a utilisé le modèle basé sur la simulation des automates. Dans presque tous les cas, les modèles mathématiques, qui sont utilisés dans la science sociale, surtout l'économie, sont des modèles de type fonctionnel. Dans tels modèles, ce qui est le plus important est la solution des équations simultanées. Mais, dans ce type des modèles, nous ne pouvons pas représenter des interactions vivantes entre les agents élémentaires. Nous devons utiliser des types de modèle automate comme Axelrod pour représenter les résultats des interactions vivantes.

3.2 Qualité 2: Axelrod rapportait inconsciemment le NNTP au gène dans l'algorithme génétique.

Axelrod a traduit les stratégies, qui sont décrites en règles de type « si-alors », sur une ligne de code génétique de 64 caractères. Puis, il a ajouté des choix aux derniers 3 jeux comme 6 caractères dans le code génétique. Nous pouvons penser que ces 6 caractères signifient aussi des règles spéciales de type « si-alors ». Cette ligne de 70 caractères représente un ensemble de règles de type « si-alors ». C'est un NNTP. Mais il ne connaissait pas la notion du NNTP. Donc Axelrod a inconsciemment utilisé le NNTP.

3.3 Défaut 1: Axelrod ne pouvait pas suffisamment saisir le sens de l'environnement social.

Dans les jeux évolutifs d'Axelrod, les environnements, à qui les agents ajuster par l'algorithme génétique, est donné en matrice des gains. Parce que c'est fixé pendant la simulation, l'environnement fonctionnerait comme si c'était l'environnement naturel. En fait, c'est l'environnement social. Les peuples de

société peuvent changer leur environnement social leur-même. Donc, Axelrod ne pouvait pas suffisamment saisir le sens de l'environnement social.

3.4 Défaut 2: Axelrod utilisait l'aléatoire dans le changement du NNTP

Axelrod a croisé et a fait mutation des individus à proportion de un pour dix. Dans le cas de croiser, il choisit une place dans la ligne du code génétique à la probabilité de croiser. Ensuite, il a coupé les codes génétique des parents à cette place. Et puis, les côtés différents des codes des parents sont copiés au nouveau bébé. Dans le cas de la mutation, il a fait une erreur consciente en copiant la probabilité de la mutation. En ces deux cas, Axelrod a utilisé un aléa au lieu de l'intelligence.

4. La méthode de l'ISEM

4.1 Changement du gène en NNTP

En les individus ayant ses intelligences, la société est évoluée par les intelligences humaines. Ces intelligences humaines, est le travail des esprits humains. L'esprit humain est un automate. Dans l'automate il y a un programme avec qui l'automate marche. Le programme dans l'automate est un « Non-Neumann Type Program » ou un « NNTP ». On peut décrit un NNTP en règles de type « si-alors ». Donc, on peut décrire le NNTP dans l'esprit humain en règles de type « si-alors ». En résolvant son problème, chacun agent de la société change son esprit. Chacun agent résout son problème par son intelligence. Et la solution de son problème devient un savoir qui est aussi un sous ensemble de son NNTP. En bref, chacun agent change son NNTP, en résolvant son problème lui-même. Ce changement du NNTP fait changer sa conduite. Ensuite, sa nouvelle conduite doit être choisie par le choix social. Le choix naturel choisit des codes génétiques par l'aptitude à l'environnement naturel. Mais, le choix social choisit des NNTPs par l'aptitude à l'environnement social. L'environnement social est aussi déterminé par le NNTP. Donc, on peut changer l'environnement social par la programmation du NNTP. Cette réflexion nous indique l'idée de la programmation du NNTP social.

4.2 Changement de l'environnement naturel en environnement social

Dans le cas du GA, des codes génétiques doivent s'adapter à l'environnement naturel. Mais, dans le cas de l'application de l'ISEM à l'évolution sociale, des NNTPs doivent s'adapter à l'environnement social. Il y a une différence essentielle entre l'environnement naturel et l'environnement social. Les individus ne peuvent pas changer leur environnement naturel. Mais, les individus peuvent changer leur environnement social. L'environnement social est aussi déterminé par le NNTP. On peut changer les environnements sociaux par la programmation du NNTP. Cette réflexion nous indique l'idée de la programmation du NNTP de l'environnement social. Les systèmes économiques sont aussi des environnements sociaux. Nous pouvons changer nos environnements nous-même.

4.3 Changement de l'aléa en intelligence humaine

Dans le cas du GA, le changement du code génétique est causé par la mutation ou le croisement. Dans les deux cas, il y a un rôle important de l'aléa. Mais, l'aléa n'est pas efficace. Il y a besoin de beaucoup de sacrifices pour l'aléa. Dans l'aléa, un peu réussites en changement du code génétique ont besoin de presque tout sacrifice. Dans le cas de l'évolution de la société humaine, il n'y a pas de place pour un tel gaspillage. Dans le cas de l'évolution sociale, au lieu de l'aléa, l'intelligence humaine joue le rôle principal. Dans le cas de la société humaine, le savoir joue le rôle du NNTP, à la place du code génétique. Les résolutions des problèmes des individus changent leurs savoirs. Ce changement du savoir permet à l'individu de changer leur conduite. En bref, les individus changent leurs conduites grâce à leurs savoirs. Les individus dans la société inventent leur nouvelle conduite à l'aide de leur intelligence. Mais, les nouvelles conduites doivent passer un test de l'environnement social. C'est le choix social. Les individus peuvent aussi prendre les choix sociaux en considération. Donc, ils peuvent éviter le rejet avec le choix social avant de pratiquer.

4.4 L'intelligence humaine peut changer l'environnement social

Parfois, les gens rencontrent leurs difficultés, dans leurs vies ordinaires. Quand ils rencontrent leurs difficultés, ils veulent connaître quels problèmes ils rencontrent. Chacune personne connaît si c'est son nouveau problème. C'est la reconnaissance du problème. Si le problème est comme ce que il a déjà connu, il peut résoudre ce problème sans augmenter son savoir. Mais, si c'est un nouveau problème, il doit commencer de rechercher la solution de ce problème. Ensuite, il invente une nouvelle solution par son intelligence. Dans ce cas, son savoir augmente, parce que cette nouvelle solution est son nouveau savoir. Dans quelque cas, le problème est commun au peuple. Donc, la personne, qui rencontre ce problème, peut trouver beaucoup de peuple qui rencontrent le même problème. Dans ce cas, il peut inventer quelque coopération avec ces peuple qui ont le même problème. Et il peut persuader ce peuple de résoudre ce problème par leur coopération. Puis, il peut changer leur environnement social et résoudre la cause du problème. Nos intelligence permet nous de changer nos environnement social nous-même. En bref, nous pouvons programmer le NNTP de nos société. Alors, la société est changé. Donc, nous pouvons programmer nos environnement social par la programmation du NNTP.

5. La possibilité du Prolog

Dans le cas de la représentation du savoir de l'agent qui est un automate, le NNTP a le style le plus courant. Le langage informatique Prolog est très utile à la représentation du NNTP.

5.1 La représentation du NNTP

Le « NNTP » signifie « Non-Neumann Type Program ». Un NNTP est un ensemble des règles qui sont décrit en type « si-alors ». Dans le modèle qui est utilisé par Axelrod, le « chromosome » est représenté par les 70 bits de 0s et 1s. Cette représentation est restreint par l'utilisation du GA. Mais, telle suite de 0s et 1s ne peut pas décrire le savoir plus compliqué à

l'environnement social. Nous avons besoin de utiliser la méthode plus compliqué pour la représentation de l'environnement social et excéder le GA. Si nous utilisons le Prolog à décrire le NNTP, nous pouvons décrire du NNTP avec structures compliqué. C'est très important pour représenter l'intelligence qui est essentiel à la méthode de l'ISEM. Il y a la méthode de l'ISEM au-dessus du GA.

5.2 La représentation de l'ISEM

En utilisant le Prolog, nous pouvons décrire le NNTP avec la structure compliqué. C'est important pour représenter l'intelligence qui est essentiel à la méthode de l'ISEM. Nous avons besoin du langage informatique plus puissant pour décrire des savoirs en règles de type « si-alors ». Parce que nous devons décrire des règles de type « si-alors » au lieu de la suite de 0s et 1s. Le langage informatique Prolog permet nous de décrire tels savoirs compliqué.

On peut représenter l'addition, la soustraction et les autre opérations du savoirs, qui sont les opérations de l'ensemble, en Prolog.

Axelrod ne pouvait pas suffisamment saisir le sens de l'environnement social. Mais, nous pouvons écrire la matrice des gains dans un jeu du dilemme du prisonnier en langage informatique du type « si-alors ».

Si nous décrivons les règles en Prolog, nous pouvons modifier les règles pendant que nous remuons les règles à l'ordinateur. C'est une propriété très utile du Prolog. Le LISP a aussi cette propriété. Mais, le Prolog est meilleur que le LISP, parce que des codes en Prolog sont les règles de type « si-alors ». Donc, nous pouvons décrire très simplement le NNTP en Prolog.

Nous ne pouvons pas décrire l'évolution social, avant nous utilisons ce langage informatique sophistiqué pour l'intelligence artificielle.

Si nous écrivons la matrice des gains en langage informatique du type « si-alors », nous ne pouvons plus l'exprimer sous la forme codée par ligne des caractères. Mais nous pouvons l'exprimer en Prolog qui est un langage de programmation logique.

$$(p_1(x) \wedge q_1(x)) \vee (p_2(x) \wedge q_2(x)) \rightarrow r(x)$$

La formule logique ci-dessus est exprimé par programme en Prolog ci-dessous.

$$r(X) : -p_1(X), q_1(X).$$

$$r(X) : -p_2(X), q_2(X).$$

En Prolog, le opérateur logique « et » noté « ^ » est exprimé par « , » et le opérateur logique « ou » noté « v » est exprimé par « le point et le retour et un autre formule posé la tête même ».

Nous pouvons décrire la matrice des gains avec des règles de type « si-alors » ci-dessous.

	choix	choix		gain	gain
	A	B		A	B
Si	C	C	Alors	3	3
Si	C	D	Alors	0	5
Si	D	C	Alors	5	0
Si	D	D	Alors	1	1

Ces règles sont décrites en Prolog, qui est un langage informatique, précédé.

```
gain_du_joueur_a(3):-
    choix_du_joueur_a(c), choix_du_joueur_b(c).
gain_du_joueur_a(0):-
    choix_du_joueur_a(c), choix_du_joueur_b(d).
gain_du_joueur_a(5):-
    choix_du_joueur_a(d), choix_du_joueur_b(c).
gain_du_joueur_a(1):-
    choix_du_joueur_a(d), choix_du_joueur_b(d).
gain_du_joueur_b(3):-
    choix_du_joueur_a(c), choix_du_joueur_b(c).
gain_du_joueur_b(5):-
    choix_du_joueur_a(c), choix_du_joueur_b(d).
gain_du_joueur_b(0):-
    choix_du_joueur_a(d), choix_du_joueur_b(c).
gain_du_joueur_b(1):-
    choix_du_joueur_a(d), choix_du_joueur_b(d).
```

5.3 La représentation de l'intelligence humaine

La représentation de l'intelligence humaine. C'est le problème principal de l'intelligence artificielle. L'intelligence humaine est l'activité de l'esprit humain. Et l'activité est essentiellement les utilisations les savoirs humains et les réflexions. Donc, la représentation du savoir humain est essentiel pour représenter l'intelligence humaine. Le NNTP est un ensemble de savoirs humains. Et nous pouvons décrire le NNTP en règles de type « si-alors ». Ensuite, la manière de l'activité de l'esprit humain est aussi décrit en règles de type « si-alors ». Puis, le NNTP est l'ensemble complet de programme qui meut l'esprit humain. Donc, nous pouvons décrire complètement l'intelligence humaine en règles de type « si-alors ». Mais, nous avons besoin de décrire cet ensemble de règles de type « si-alors », qui a une structure très compliqué. Donc, le langage informatique Prolog est nécessaire. Ensuite, en utilisant le Prolog, nous pouvons approcher de décrire l'intelligence humaine. La méthode de l'ISEM a besoin de décrire l'intelligence humaine. Peut-être cette méthode va être le plus important genre de la recherche dans l'intelligence artificielle.

Référence

[Axelrod1997] Robert Axelrod: *The complexity of cooperation: agent-based models of competition and collaboration*, Princeton University Press, 1997.

[Axelrod2006] Robert Axelrod: *Comment réussir dans un monde d'égoïstes: Théorie du comportement coopératif*, Odile Jacob, 2006.

[Bratko1986] Ivan Bratko: *Prolog Programming for Artificial Intelligence*, Addison-Wesley, 1986.

[Holland1992] John H. Holland: *Adaptation in Natural and Artificial Systems*, The MIT Press, 1992.

[Mado2005] Shinichiro Mado: 'Agents and non-Neumann Type Program', The 19th Annual Conference of the Japanese Society for Artificial Intelligence, 2005.

[Mado2006] Shinichiro Mado: 'Agents' Evolution and Ability to Solve Problems', The 20th Annual Conference of the Japanese Society for Artificial Intelligence, 2006.