人工知能学会第二種研究会資料
汎用人工知能研究会
SIG-AGI-021-04

報酬最大化ＡＧＩのための意思疎通機構の設計とプロトタイプ実装

Design and prototype implementation of a communication mechanism for reward-maximizing AGI

一杉裕志[1*]　中田秀基[1]　高橋直人[1]　竹内泉[1]　佐野崇[2]
Yuuji Ichisugi[1]　Hidemoto Nakada[1]　Naoto Takahashi[1]　Izumi Takeuti[1]　Takashi Sano[2]

[1] 産業技術総合研究所 人工知能研究センター
[1] National Institute of Advanced Industrial Science and Technology (AIST), AIRC
[2] 東洋大学 情報連携学部情報連携学科
[2] Faculty of Information Networking for Innovation And Design, Toyo University

Abstract: We designed and implemented a prototype mechanism for agents to communicate with each other for the purpose of reward maximization. The agents can be interpreted as solving POMDP in an approximate manner, and are designed to use their actions, inferences, and communication in a purposive manner. We examined the validity of the design by writing a test program that runs on the prototype implementation. This mechanism is also a candidate for a computational model of human communication.

Alice はハサミの場所が知りたいと思う。
Alice はハサミの場所を誰が知っているか考える。
Alice は Bob が知っていると思い出す。
Alice は「Bob, ハサミはどこにある？」と聞く。
Alice は返事を待つ。
Bob は「ハサミは Room1 にある」と返事する。
Alice はハサミは Room1 にあると知る。

図 1: テストプログラムの対話シナリオ

1　はじめに

自然言語を通じて人間と自在に意思疎通できる機械を作ることは、いまだ達成されていないＡＩ研究者の目標の１つである。それは有用な汎用人工知能（AGI）が当然持つべき機能でもある。

我々は再帰的強化学習・生成モデル・プログラム合成の３つを中核技術とした脳型ＡＧＩアーキテクチャの実現を目指している [1]。その一環として、プログラム合成対象言語 Pro5Lang を設計した [2]。Pro5Lang は、再帰的強化学習 RGoal [3] を用いてエージェントの報酬を最大化するプログラムを合成するための、合成対象言語である。

*連絡先：産業技術総合研究所
茨城県つくば市梅園１－１－１ 中央第１
E-mail: y-ichisugi@aist.go.jp

本稿ではプログラミング言語 Pro5Lang に追加した意思疎通機構について述べる。ここでは言語活動を含む様々な知能は報酬最大化という目的から生じるという仮説 [4] を前提としている。将来は、発話計画・言語理解を行う Pro5Lang のプログラムをエージェント自身が自らの経験や他者との対話等を通じて自律的に獲得することを目指している。今回のプロトタイプ実装は、発話計画・言語理解のプログラムを人間が手で書いて、設計の妥当性を検証する。

本稿は以下のような構成になっている。2 節で提案機構について述べ、3 節でテストプログラムとその動作について説明する。4 節ではテストプログラムの拡張性について考察し、5 節で関連研究との関係について述べる。最後に 6 節でまとめを述べる。

2　提案機構

2.1　Pro5Lang の概要

Pro5Lang は論理型言語（数理論理学を基礎にしたプログラミング言語）と機械語（コンピューターを構成する論理回路が直接解釈実行できる言語）の特徴を合わせ持ったプログラミング言語である。情報の記憶場所としてレジスタと連想記憶装置を有し、そこに証明済みの命題を記憶する。

```
g1 = <ハサミは PLS にある>
1: rule(__, g1, call(<ハサミがどこにあるかを PLS が知っている>))
2: rule(<ハサミがどこにあるかを z が知っている>, g1, set(<自分は z にハサミがどこにあるか聞きたい>))
3: rule(<自分は z にハサミがどこにあるか聞きたい>, g1, SayIt)
4: rule(<自分は z にハサミがどこにあるか聞いた>, g1, Listen)
5: rule(<__ が自分に「ハサミは x にある」と言った>, g1, set(<ハサミは x にある>))
```

図 2: 図 1 のシナリオにしたがって会話する Alice の行動ルール集合の疑似コード。行動ルールは rule(状態, サブゴール, 行動) という形式で定義される。“PLS” は unknown 以外の値、 “__” は任意の値にそれぞれマッチする。

Pro5Lang のプロトタイプシステムを、Java 言語のソースコード中に埋め込める DSL (Domain-Specific Language) として実装し、この DSL で様々なテストプログラムを手で書いて動かしてみることにより、処理系の動作検証と言語仕様の妥当性の検証を進めている。

プログラムは行動ルールの集合として定義される。行動ルールは rule(s,g,a) という形をしており、エージェントは、現在のレジスタの状態 s およびサブゴールの値 g とマッチするパターン (s,g) を持つ行動ルールを非決定論的に 1 つ選択し、その行動 a を実行する。行動ルール集合は強化学習における行動価値関数を圧縮表現したものであり、行動ルールの価値の学習が進めばエージェントは報酬を最大化するように合理的に振舞うようになる。

エージェントが取り得る行動のうち、call, set, recall 命令はそれぞれサブルーチン呼び出し、レジスタの値の更新、宣言的知識の想起を実行するプリミティブである。

Pro5Lang の詳細については [2] を参照されたい。

2.2 意思疎通機構

意思疎通の本質部分に注力するため、エージェントどうしは音声や文字列ではなく、発話内容の内部表現を直接やりとりするように実装することとした。各エージェントは他者の発話内容を一時的に保持する**文バッファ**を持ち、話し手は聞き手の文バッファに伝えたい内容を直接書き込む[1]。

意思疎通は Listen と SayIt という 2 つのプリミティブによって行われる[2]。

- Listen は相手の発話を待つ命令である。文バッファに値がまだなければ、何もしない。文バッファに値があればその内容を a レジスタに移動する。
- SayIt は発話準備した内容を実際に発話する命令である。a レジスタに置かれている発話準備内容を発話相手の文バッファに書き込み、自分の a レジスタには自分が発話した事実を表す命題を入れる。

[1] 以前に [2] で発話内容を s レジスタに保持する予定であると説明したが、レジスタとは別の場所に保持することとした。

[2] ヒトの脳内では前運動野に相当する下位の階層が実行すると想定している。

3 テストプログラム例

図 2 は、図 1 のシナリオにしたがって会話する Alice の行動ルール集合の疑似コードである。

行 2 は発話の準備を行う行動ルールであり、実際の DSL のコードは以下のようになる。

```
rule(w(a(ThatO1, c(Now, Here, z, Knows, 0, 0),
                 c(Now, Wh, Scissors, Exists, 0, 0))),
     g1,
     set(a(ThatO2, c(Now, Here, Alice, WantsToSayTo, z, 0),
                   c(Now, Wh, Scissors, Exists, 0, 0))));
```

命題は複文程度の情報を表現することができ、各節は c(いつ, どこで, 誰が, 何をした, O1, O2) という形式で記述する。接続詞 ThatO1, ThatO2 はそれぞれ主節の O1, O2 に従属節の内容が埋め込まれていることを表す。Wh は、「どこに」「何を」などを表すシンボルである。現状、代名詞は実装されていないので自分自身を固有名詞 Alice で表現している。

行 3 と行 4 は準備した内容の発話と、そのあとの応答を待つ行動ルールであり、実際のコードは以下のようになる。

```
rule(w(a(ThatO2, c(Now, Here, Alice, WantsToSayTo, z, 0),
                 c(Now, Wh, Scissors, Exists, 0, 0))),
     g1,
     SayIt);
rule(w(a(ThatO2, c(Now, Here, Alice, SaidTo, z, 0),
                 c(Now, Wh, Scissors, Exists, 0, 0))),
     g1,
     Listen);
```

発話が完了すると a レジスタの WantsToSayTo が SaidTo に自動的に変化する。行 4 の実行 (Listen) は、返答が来るまで繰り返される。

行 5 は Bob が応答し、 a レジスタに応答内容が書き込まれた後に実行される行動ルールで、実際のコードは以下のようになる。

```
rule(w(a(ThatO2, c(Now, Here, __, SaidTo, Alice, 0),
                 c(Now, x, Scissors, Exists, 0, 0))),
     g1,
     set(a(Now, x, Scissors, Exists, 0, 0)));
```

Bob が「ハサミは Room1 にある」と言ったら、Alice は < ハサミは Room1 にある > という命題が証明されたと解釈し、その命題を a レジスタに set する。その値は自動的に連想記憶装置にも加えられ、そのあとの情報処理に利用することができる。

このテストプログラムでは、 a レジスタの値が対話シナリオの進行を制御する重要な役割を果たしている。プログラムの書き方によっては同じ推論や同じ質問を何度か繰り返すという無駄な動作が発生するが、このプログラムでは行動ルールの選択タイミングが適切に記述されているため無駄な動作は生じない。

4 テストプログラムのシナリオの拡張性

提案機構は単純なルールベースの対話システムとは異なり、発話計画も発話理解も報酬最大化を目的とするプログラムの実行として行われるため、その振る舞いは柔軟かつ合目的的である。図 2 の疑似コードは以下の拡張性を持ち、様々な状況に柔軟に対応できるエージェントの振る舞いが記述可能になると考えられる。強化学習による学習が進めば、ここで述べる多様な選択肢のうち、高い価値を持つものが選択されるはずである。

行 1 は誰かに聞くことを前提で誰に聞くかを考える行動ルールだが、それ以外の選択肢、例えば自分で考える、自分で探す、などの行動ルールを追加することができる。

行 2 の発話準備内容は「ハサミはどこ？」という直接的な質問になっているが、これ以外にも状況に応じて「ハサミちょうだい」「ハサミの場所知らない?」「誰かハサミを使った？」などの聞き方を検討し、もっとも妥当な聞き方を選択することが可能である。

行 5 は相手の返答への対処だが、相手は「ハサミは Room1 にある」という直接的な応答をするとは限らず、単に「Room1」と答えたり、「きのう Room1 で見た」という答え方をすることもあり得る。また、 Alice の予想に反して Bob はハサミの場所を知らず「自分は知らないけど Carol が知っている」と答えるかもしれない。あるいは「今は使わせたくない」と答えるかもしれない。このようなさまざまな返答に対し、機械的に処理するのではなくまず相手の意図を推論し、その上で適切に対応する行動ルールを用意しておくことができる。

行 5 はまた、返答の内容を素直に信じる行動ルールになっているが、状況によっては「相手の発言は間違っているかもしれない」「相手はうそをついているかもしれない」といった他の解釈を検討すべき場合もある。そのような対応を記述することも可能である。

5 関連研究

5.1 部分観測マルコフ決定過程

部分観測マルコフ決定過程 (POMDP) は、環境のモデルが与えられていれば、信念状態（エージェントが推定する環境の状態の確率分布）を状態とみなして belief MDP [5] として定式化しなおすことで、 MDP を解く様々な手法で解けるようになる。POMDP を解いたエージェントは、必要な情報を得るために環境を能動的に観測したり、他者に聞くという行動を取るようになる。

Pro5Lang は、belief MDP を近似的に解いていると解釈することができる。変数は unknown という値を取ることができる。ある変数 v の値が unknown であるということは、エージェントが持つ信念状態において変数 v の周辺分布がほぼ一様分布であることを意味している。したがって Pro5Lang の行動ルール選択は、環境の状態に対するパターンマッチではなく、実は信念状態の近似表現に対するパターンマッチにより行われる。

Pro5Lang と belief MDP には違いもある。隠れた状態の推論は belief MDP では環境モデルを用いて瞬時に行われるかのように定式化されるが、 Pro5Lang は隠れた状態の推論も「脳内アクション」として数ステップかけて実行される。サブゴール達成に無関係な状態の推論は行わないので効率的であるが、一方で、必要なはずの情報を誤って無関係だと見なし、推論しない可能性もある。これは欠点だが、もしヒトの脳も同じであるなら脳型 AGI としては許容できる性質である。

対話システムを POMDP に基づいて構築することによる利点は [6] で詳しく論じられている。しかしそこではスケーラビリティの問題も指摘されている。Pro5Lang で記述される対話機構は、再帰的強化学習 RGoal のサブタスク共有・時間抽象・状態抽象の 3 つの恩恵により、大規模でも現実的なコストで解けるようになると思われるが、今後より大規模なテストプログラムを動かすことで実証していく予定である。

5.2 関連性理論

言語学における語用論に属する理論の 1 つに関連性理論 [7] がある。関連性理論は発話がいかに理解されるかということに関する理論であり、話し手は聞き手にとって有益なことを伝えようとしているという前提の下で、聞き手は話し手の意図を推論すると考える。関連性理論は数学的に定式化されておらずそのまま計算機上に実装することはできない。しかし、単純化した状況における指差し行動の関連性を効用と結び付け、POMDP に基づいて定式化する試み [8] がある。

Pro5Lang による言語理解は報酬最大化を目的として実行されるため、関連性理論との親和性は高い。また、関連性理論では発話理解は機械的に行われるのではなく、聞き手は発話者の意図を推論すると考えるが、これは、報酬を最大化するために隠れた状態を推論するという Pro5Lang の動作と整合性があるように思われる。今後、より複雑な環境で Pro5Lang を用いた様々な対話シナリオを実装することで、関連性理論との関係がより明らかになるとともに、Pro5Lang が計算機で実行可能なヒトの意思疎通の計算論的モデルになると期待できる。

5.3 Inquisitive semantics

疑問文の意味を形式的に表現する方法の１つに inquisitive semantics [9] がある。質問者の意図は、質問に対する答えの選択肢の集合として定式化される。Pro5Lang では疑問文の発話意図は “PLS” を含むサブゴールで表現される。この表現方法は inquisitive semantics よりもはるかに表現力が劣るが、別途サブルーチンを用いれば任意の疑問文の発話意図が表現可能になるだろうと考えている。Pro5Lang のサブゴールの表現力の制約がヒトの発話の特性と一致しているかどうかを検証することができれば、 Pro5Lang の意思疎通機構の計算論的モデルとしての妥当性を示すことができるだろう。

6 まとめ

報酬最大化を目的として動作するエージェントどうしが、お互いに意思疎通するための機構を設計し、プロトタイプ実装を行った。現在は単純なテストプログラムしか動いていないが、今後エージェントの動作環境を拡張し、複雑な対話シナリオをデモすることを目指す。

Pro5Lang の表現力を増すための拡張もまだ必要で、少なくともタイムアウト、割り込み、メモリへのポインタの機構の追加が必要だと考えている。効率的な実行のためにはニューラルネットのような非記号的な情報処理による補助も不可欠だろう。

提案機構では発話計画や意味理解は作り付けの機構ではなくプログラムの実行によって行われるため柔軟性が高く、また、対話履歴や意味理解の結果は証明済み命題として蓄積されて次の発話や意味理解に役立てることができるため、将来的には非常に内容豊かな会話が実現されることが期待できる。

謝辞

本研究は JSPS 科研費 JP22K12188 の助成を受けたものです。

参考文献

[1] 一杉裕志. 報酬最大化原理にもとづく脳型ＡＧＩアーキテクチャの構想. 第 18 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2021.

[2] 一杉裕志, 中田秀基, 高橋直人, 竹内泉, 佐野崇. 汎用人工知能のためのプログラム合成対象言語 Pro5Lang のエピソード記憶機構. 第 20 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2022.

[3] Yuuji Ichisugi, Naoto Takahashi, Hidemoto Nakada, and Takashi Sano. Hierarchical reinforcement learning with unlimited recursive subroutine calls. In *Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning*, pp. 103–114, Cham, 2019.

[4] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. *Artificial Intelligence*, Vol. 299, p. 103535, 2021.

[5] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, Vol. 101, pp. 99–134, 1998.

[6] Jason D Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, Vol. 21, No. 2, pp. 393–422, 2007.

[7] Dan Sperber and Deirdre Wilson. *Relevance: Communication and Cognition*. Blackwell, 1986.

[8] Kaiwen Jiang, Annya L. Dahmani, Stephanie Stacy, Boxuan Jiang, Federico Rossano, Yixin Zhu, and Tao Gao. What is the point? a theory of mind model of relevance. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 44, 2022.

[9] Ivano Ciardelli, Jeroen Groenendijk, and Floris Roelofsen. *Inquisitive semantics*. Oxford University Press, 2018.