

# 不確実性に対処可能な Teleo-reactive Program の意味論とその応用

後藤 勇樹<sup>1,a)</sup> 藤田 恵<sup>2</sup> 新出 尚之<sup>3</sup>

**概要:** 我々はこれまでの研究で、現実世界の動的環境に対応可能なロボットの自律的制御方法を、Teleo-Reactive Program を基盤として確立してきた。実験結果から、ロボットが時に予想外の行動を選択する場合でも、最終的に目標を達成する事が確認できたが、この事実を反映する有効な意味論は得られていなかった。本論文ではまず、我々の方法に対する意味論を構成する事により、目標の達成可能性を示す Model check の方法を示す。また、その際に予想外の行動を行う不確実性に対し、我々の方法でどの様に対処しているかを解説する。次に、今回の意味論の応用について述べる。具体的には、過去の出来事に対して合理的な解釈を行う事で、次の行動方針を決定するエージェントの構想を述べる。

## A semantics of Teleo-reactive Program that can handle uncertainty and its application

GOTO YUKI<sup>1,a)</sup> FUJITA MEGUMI<sup>2</sup> NIDE NAOYUKI<sup>3</sup>

**Abstract:** Our past research has been establishing a method of autonomous control of robots which can respond to dynamic changes of the real worlds. Though the robots can eventually achieve their goals even if they occasionally select the unexpected behaviors, we did not have any effective semantics which reflects that fact yet. In this paper, we construct a semantics for the autonomous control method of real-world robots which is shown in our past research. Then, based on it, we present a model-checking method to show the possibility of achieving a goal. In addition, we describe our method to handle the uncertainty of the real world. Next, we show the application of our semantics. Especially, we propose an idea of agent system which, by giving rational interpretations to past events, determines a policy of selecting its actions.

**Keywords:** Real world, Autonomous control, Uncertainty, Teleo-Reactive Program

### 1. はじめに

ロボットの様な実世界で活動するエージェントについて、これまで幾つかの知的制御方法が考案されてきた。特に、Nilsson によって 1992 年に考案された Teleo-Reactive(T-R) Program[1] は、動的環境による外部知覚の連続的な変化を取り入れたエージェントシステムであるが、今までに十分な知能ロボティクスへの応用をされてきたとは言い難

い。この背景には、T-R Program は連続的な環境を扱っている為、通常の離散的なプログラム意味論ではその本質を捉えきれず、最終的な目標達成能力を保証する事が出来なかった事が挙げられる。

例えば、Kowalski は T-R Program を元に Logic based T-R Program(LTR) を構成し、その操作的意味論を与える事で目標達成能力を保証している [2] が、その方法は連続的な環境の動的変化を扱えるものではなかった。

また、我々も過去に、LTR と同等の論理プログラムを用いた、自律的移動ロボットの知的制御の研究を行った [3] が、その時にも、そのプログラムの目標達成能力の保証を

<sup>1</sup> 京都大学数理解析研究所

<sup>2</sup> 奈良女子大学大学院人間文化研究科

<sup>3</sup> 奈良女子大学大学院自然科学系情報科学領域

<sup>a)</sup> gotoyuki@kurims.kyoto-u.ac.jp



(1) 初期状態



(2) Qbo がターゲットを発見し  
 その方向へ進んでいるところ



(3) Qbo が障害物を発見したところ



(4) Qbo からターゲットが見えない状況



(5) Qbo が障害物を回避



(6) Qbo が再度ターゲットを発見



(7) Qbo がターゲットの方向へ進む



(8) Qbo がターゲットの前に到達

図 1 我々の以前の実験

する事が出来なかった。

本論文では、我々の用いたプログラムを LTR の形で解釈し直し、その意味論を実験環境を模した位相空間上に展開する事によって、エージェントの目標達成能力の保証を行う。また、この際に、実世界環境がエージェントに与える不確実性に関しても考察を行う。最後に、この意味論の応用方法として、過去の出来事に対して合理的な解釈を行う事で、次の行動方針を決定するエージェントを考察する。

## 2. 実験時の LTR と実験の様子

### 2.1 T-R Program と LTR

通常の T-R Program の形式的な記述は、以下の様になる。

$$K_1 \rightarrow a_1$$

$$K_2 \rightarrow a_2$$

$$K_3 \rightarrow a_3$$

...

$$K_m \rightarrow a_m$$

ここで、各  $K_i$  は条件を、各  $a_i$  は行動を表現している。エージェントは常に連続的な外部知覚を行っており、その情報を元に、条件  $K_1, K_2, \dots$  を上から順に評価し、最初に満たされた条件  $K_j$  と結び付いた行動  $a_j$  を行う。通常は最後の条件  $K_m$  には *TRUE* を定め、常に何らかの行動を行うように構成する。行動中のエージェントもその外部知覚の変化に伴い、上記の分岐条件を利用して動的に行動を切り替える。

T-R Program の利点は、動的な環境変化が起きた時に自然と適切な行動が取れる点にある。しかしこの方法は、外部環境の知覚情報を連続的に取得する必要があり、実装するには論理回路のようなものを用いる必要がある。

これに対し、Kowalski は LTR を考案する際に、外部環境を観測する述語を明示的に導入し、観測の度に行動を行う方針を明らかにした。この手法だと、通常の論理プログラムの方法で書ける為、実装も容易である。我々も、次節で紹介する様に外部環境を観測する基本行為を構成したが、LTR より明示的に知覚行為と基本行為を交互に実行す

る方針を取っている。

## 2.2 我々の実験とその制御方法

我々は以前、移動ロボットを用いた知的制御の実験を行った。実験に用いたのは TheCorpora 社製の Qbo Pro Evo (以下 Qbo と呼ぶ) というロボットで、大きな2つの車輪の前に障害物判定が可能な超音波センサーがあり、また、目の位置にあるカメラで物体認識が可能である。

実験の目標は予め置かれたターゲットの前まで移動する事であるが、Qbo は事前にターゲットの位置情報を持たず、物体認識を行ってターゲットの存在を知る必要がある。但し、ターゲットの存在する大体の方向のみ初めに与えている。また、Qbo がターゲットを発見した後で、ターゲットが見えなくなる大きさの障害物を目の前に置いて進路を邪魔した。図1に、この実験の様子を示す。

この時の知的制御方法は、以下の様な擬似コードで書ける。この書き方は LTR を論理プログラムで書く方法とほぼ一致する。

Goal : - *looking\_Qbo(Direction)*, Goal'.

Goal' : - *found\_Obstacle,!,*  
*search\_Qbo(Direction)*, Goal.

Goal' : - *not\_found\_Obstacle,!,*  
*forward\_Qbo(Direction)*, Goal.

コード中の *found\_Obstacle*, *not\_found\_Obstacle* は、超音波センサーを利用して、Qbo の目の前に障害物があるか否かを判定した結果を与える。この情報は、各基本行為の動作後に更新される。また、*looking\_Qbo/1*, *forward\_Qbo/1*, *search\_Qbo/1* は基本行為を表しており、大体以下のような行動を行う。

- *looking\_Qbo(Direction)*  
物体認識を行い、ターゲットが発見された場合はその方向を、発見されなかった場合には初めに与えた大体の方向を、引数 *Direction* に単一化する。
- *forward\_Qbo(Direction)*  
引数 *Direction* に指定された方向に向かって前進する。但し、途中で障害物が目の前にきた場合や、一定の距離を進んだ場合には前進を停止する。
- *search\_Qbo(Direction)*  
引数 *Direction* に指定された方向にできる限り近くなるように、目の前に障害物が存在しない方向を探索する。その後、その方向に前進する。但し、途中で障害物が目の前にきた場合や、一定の距離を進んだ場合には前進を停止する。

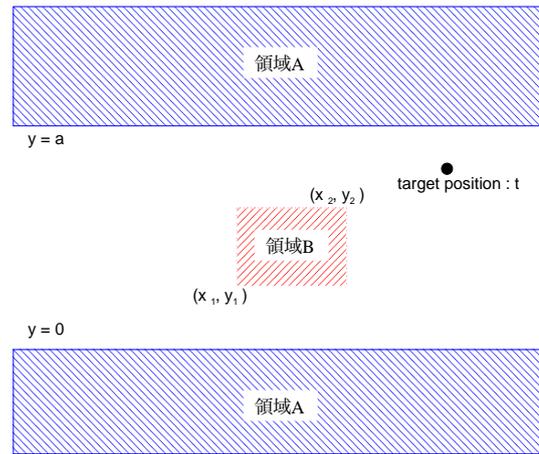


図2 実験環境のモデル

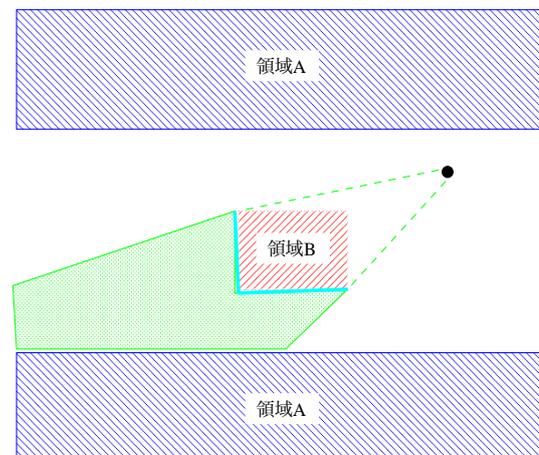


図3 LTR では、環境に従って次の行動が一つに定まる。ここでは、エージェントが居る位置によって次の行動が決定する。水色の領域で *search\_Qbo*、黄緑色の部分で *forward\_Qbo(Initial\_Direction)*、それ以外で *forward\_Qbo(target\_Direction)* を次の行動とする。

## 3. 意味論

実験結果から、我々の方法が目標達成能力を持つことを示す為には、以下に示す2つの不確実性を考慮する必要がある。

- (1) ロボットの力学的な性質や周辺環境の影響により、ロボットの動作が理想的なものからはずれてしまうという、基本行為の不確実性
- (2) ロボットの外部知覚が精確でなかったり、障害物の存在を感知出来なかったりといった様々な因子が影響して、誤った意思決定を行ってしまうという、意思決定の不確実性

1番の不確実性は、基本行為の設計時に影響した。例えば、引数で指定された方向に向く動作を行う際に、内部モーターの回転を微調整する事が出来ず、少しずれた方向に向くことになる。

2番の不確実性は、実験中に起きたものである。事例と

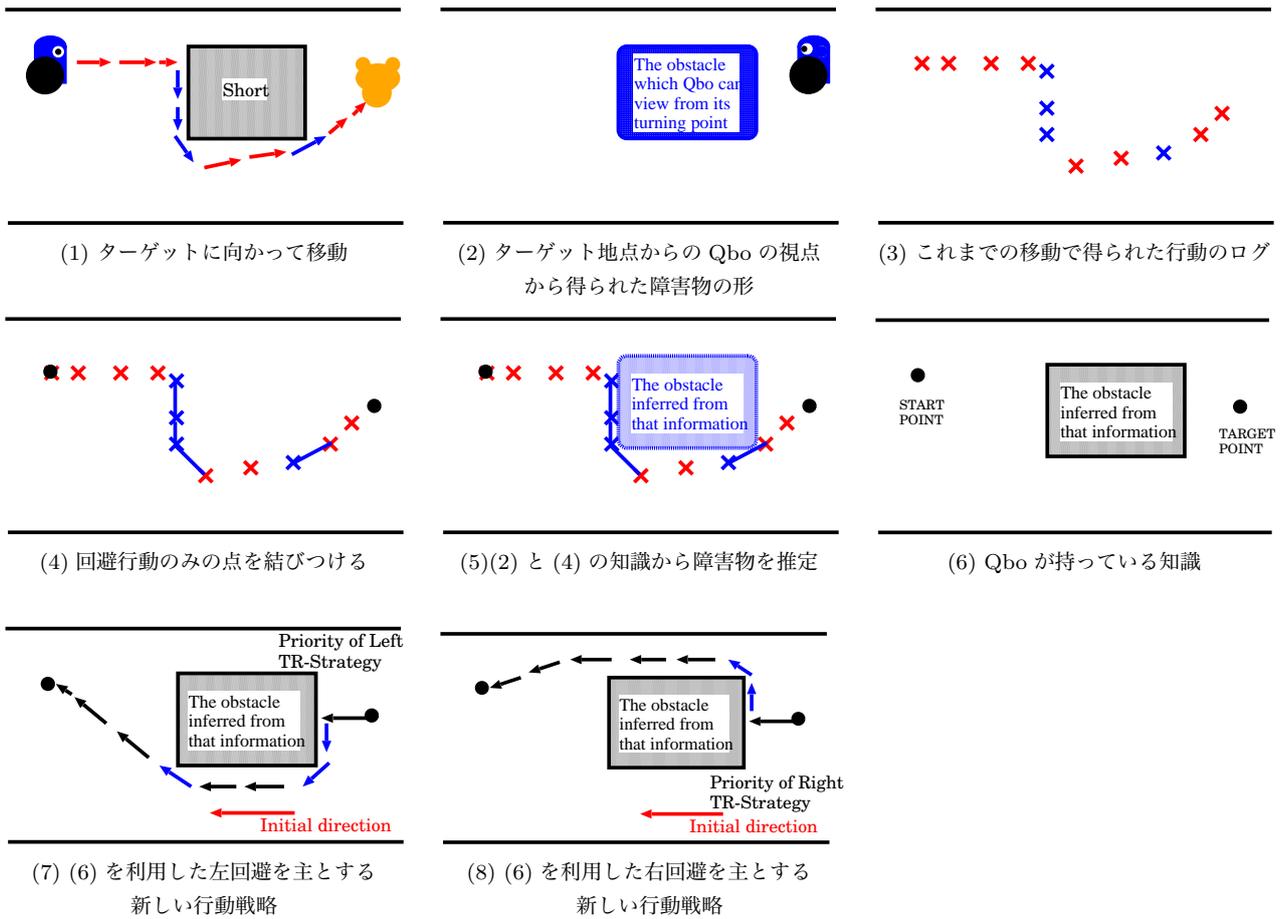


図 4 Qbo の行動から得られた知識における行動戦略の変更の流れ

しては、物体認識方法が持つ不確実性によって、柱の方向にターゲットが存在するという誤認識が発生し、その方向に向かうという誤った意思決定を行っている。

これらの不確実性に対応しつつ、目標達成能力を示す為には、先ず、理想的な行動がどのようなものを定義し、その行動から少し外れても、そこから理想的な行動を行うことで、最終的な目標達成を可能にしている事を示す必要がある。

以下では、実験環境、基本行為、不確実性を位相空間のモデルで表現し、その上で、不確実性が現れてもその度合いは基本行為を通じて減少する事を示す。最後に、エージェントの目標達成能力について示す。但しここでは、実際の実験と比べて単純な形でモデル化を行っている。単純にした部分については、後に議論を行う。

### 3.1 目標達成能力の証明

**定義 3.1 (実験環境)** 実験環境  $\mathcal{E}$  を位相空間  $(\mathbf{E}, \mathcal{O})$  で定める。但し、 $\mathbf{E}$  は、 $\mathbf{R}^2$  をユークリッド空間として、領域  $A = \{(x, y) | y \leq 0 \text{ or } a \leq y\}$  と領域  $B = \{(x, y) | x_1 \leq x \leq x_2, y_1 \leq y \leq y_2\}$  によって、 $\mathbf{E} = \mathbf{R}^2 \cap A^c \cap B^c$  で定める。また、 $\mathcal{O}$  は通常の距離位相の範囲を  $\mathbf{E}$  に限定した、相対位相とする。

**定義 3.2 (基本行為)** 基本行為  $F(Dir), S$  を共に実験環境中の写像  $\mathcal{E} \rightarrow \mathcal{E}$  として定める。具体的には、点  $p = (x, y)$  から  $Dir$  方向に一定距離  $d$  進んだ点を  $q$  として、

$$F(Dir)(x, y) = \begin{cases} t, & \text{if } \overline{pq} \cap B = \phi \text{ and } t \in \overline{pq} \\ q, & \text{elseif } \overline{pq} \cap B \neq \phi \\ r, & \text{else} \end{cases}$$

$$S(x, y) = \begin{cases} (x, y + d), & \text{if } y + d \leq a \\ (x, a), & \text{else} \end{cases}$$

と定める。但し  $r$  は条件下で、 $B$  の境界と線分  $\overline{pq}$  の交点の内、最も点  $p$  に近い点とし、ターゲットの位置を  $t$  と置く。以下では、基本行為全体の集合を  $\mathcal{M}$  と置く。

**定義 3.3 (分岐条件)** 今、エージェントの現在地を  $s \in \mathcal{E}$  と置く。この時、エージェントの次の行動は以下の様な分岐条件  $\mathcal{I}: \mathcal{E} \rightarrow \mathcal{M}$  で決定される。

- (1)  $s$  が  $B$  の境界上にある場合、基本行為  $S$  を行う。
- (2) 上記以外で  $\overline{st} \cap B \neq \phi$  が成り立つ場合、引数  $Dir$  に  $Init\_Dir$  を代入し、基本行為  $F(Dir)$  を行う。
- (3) 上記以外の場合、引数  $Dir$  に  $s$  から  $t$  へ向かう方向を代入し、基本行為  $F(Dir)$  を行う。

定義 3.1 中の領域  $A$  は壁のある位置を表し、領域  $B$  は障害物

を表している。定数  $a, (x_1, y_1), (x_2, y_2)$  は、適切に実験環境を表すように定めておく。実験環境の表現を図 2.2 に示す。定義 3.2 において、 $F(Dir)$  は基本行為  $forward\_Qbo/1$  を、 $S$  は基本行為  $search\_Qbo/1$  を意味している。 $search\_Qbo/1$  の引数を削ったのは、モデルの単純化の為である。定義 3.3 の構成方法は、前の章で示した擬似コードと同じ順番になっている。以下では、ターゲットの位置  $t$  は、エージェントの初期位置より  $x$  軸の正の方向に存在する事を仮定し、この時の  $Init\_Dir$  は  $x$  軸正方向を表すとす。

**定義 3.4 (不確実性)** エージェントが不確実な行動を行うとは、定義 3.3 で定まるのと別の基本行為を行うか、基本行為の実行時に定義 3.2 で定まる場所と異なる場所に移動する事である。但し、移動距離は一定距離  $d$  以下とする。

**定理 3.1 (目標達成)** エージェントが定義 3.3 に従って行動決定を行い続ける限り、途中で数回不確実な行動を行ったとしても、最終的にターゲットの位置  $t$  に到達する。

**補題 3.1 (不確実性の単調減少)** 実験環境中の連結空間  $P \subset E$  内部の各点に対して、定義 3.3 で定まる基本行為の写像を行い、その像を  $Q$  とする。

この時、 $P$  の面積  $\geq Q$  の面積 が成り立つ。

**証明 3.1 (補題 3.1)** 連結空間を実行される基本行為毎に分けて、考えれば良い。まず、基本行為  $S$  は壁に向かっての平行移動のみからなる為、面積は等しいか減少する。次に、基本行為  $F(Dir)$  で  $Dir = Init\_Dir$  の場合は、障害物にぶつかる可能性のある前進移動なので、やはり面積は等しいか減少する。最後に、基本行為  $F(Dir)$  で  $Dir$  がターゲット方向を指す場合は、一点に向けた縮小写像になる為、面積は減少する。(証終)

**証明 3.2 (定理 3.1)** 数回の不確実な行動を経て、今後は確実な行動のみを行う状態を考える。この時、実験環境下のどの位置にいたとしても、基本行為  $S$  や  $F(Init\_Dir)$  は高々数回しか行われず、残りは  $Dir$  がターゲット方向である  $F(Dir)$  が行われる。補題 3.1 の証明で触れた様に、この基本行為は縮小写像の為、実行後にエージェントが存在できる領域は確実に減少する。その為、最終的にこの基本行為の不動点に辿り着く。今、この  $F(Dir)$  の不動点は定義から、ターゲット位置  $t$  のみである為、エージェントは最終的にターゲットの位置に到達する事が分かる。(証終)

### 3.2 単純化の影響

前節ではモデルの構成の為に、実験時と比べて幾つかの単純化を行っている。例えば、基本行為  $Search\_Qbo/1$  から引数を取り除いた為、単純に障害物に沿って左に避ける行動としてモデル化されている。しかし、定理 3.1 の証明時に重要であったのは、ターゲット方向へ向かう  $F(Dir)$  のみである為、他の部分の単純化は、目標達成能力の証明の際に特別な影響を与えないものと思われる。

但し、今回はこの基本行為についても単純化している。

具体的には、実際には不可能な程の遠い距離からも、ターゲットを認識可能であるとしてしまっている。これは精確に基本行為をシミュレート出来ていない為に起きているが、今回の様に不確実性をうまく考慮する事によって、モデル化におけるこの手の問題に対処できる可能性がある。

注意すべき点として、元のプログラムは動的な環境変化にも対応可能な手法であるのに対し、今回のモデル化は静的な環境のみを考察している事が挙げられる。

## 4. 応用

以下に示すのは、今回の提案手法を元にして、より賢い動きをする為のアイデアである。例えば、 $Qbo$  がターゲット地点から元の位置に戻るというシナリオを想定する。(図 4)

(1) のようにターゲットに向かって移動する際、 $Qbo$  は自身が行った基本行為のログをとり、これとともにその行動をおこなった座標も記憶しておくことにする。(図中では、赤矢印でターゲットに向かう移動を、青矢印で回避行動を表している)そして、無事ターゲットに辿り着いたとして、ここから  $Qbo$  は元の位置に戻る体勢に入る。その時、 $Qbo$  から見える物体についても確認しておく(2)。これによって、 $Qbo$  は自身が過去に回避行動を行った時に、一体どのような物体を回避していたのか SVM によって物体検知することが可能となる。

ここから、 $Qbo$  が得た物体の画像による形状判断、これまでの行動で得られた座標のログ(3)での回避行動のみを結びつけたことによる障害物の大まかな座標(4)を元に(5)のような障害物の推定を行う。

これによって  $Qbo$  の持っている知識は(6)のようになる。

そこで、 $Qbo$  は自身の行動戦略を変更することが出来る。例えば、(7)のように回避行動を主に左から行うことによって行動できるように T-R program の規則の並べ替えにより実現可能となる。右優先に関しても同じ方法で変更可能である。

このように、論理プログラムによるロボット制御では、そのルールの並べ替えによってロボットのとり行動を大きく変更することが容易に可能である。

また、ロボットの持つ信念が動的に変更されることによってロボット自体の行動をより最適化することが容易に可能である。

## 5. まとめ

今回、不確実性に焦点を当てる事によって、LTR と同等の我々のプログラムに対して、目標達成能力の保証を行うことが出来た。こうしたロボットの知的制御方法についての Model check は、比較的珍しい研究なのではないかと思われる。

また、今回の方法を発展させて、Computational Tree

Logic 等を用いて動的環境を再現し、とある論理式で記述される環境下では目標を達成すると言ったような事が示せれば面白いのではないかと考えている。

#### 参考文献

- [1] Nilsson, N., Toward Agent Programs with Circuit Semantics, Technical Report STAN-CS-92-1412, Stanford University Computer Science Department, 1992.
- [2] Robert A. Kowalski and Fariba Sadri. 2012. Teleo-Reactive abductive logic programs. In Logic Programs, Norms and Action, Alexander Artikis, Robert Craven, Nihan Kesim Çiçekli, Babak Sadighi, and Kostas Stathis (Eds.). Springer-Verlag, Berlin, Heidelberg 12-32.
- [3] Fujita, Megumi., Goto, Yuki., Nide, Naoyuki., Satoh, Ken., Hosobe, Hiroshi, An Architecture for Autonomously Controlling Robot with Embodiment in Real World, Submission for proc. of KRR-ICLP2013, 07/2013