推論規則の価値を階層型強化学習 RGoal を用いて学習する手法の提案

汎用人工知能研究会
2020-02-20

一杉裕志[1*]　中田秀基[1]　高橋直人[1]　佐野崇[2]

Yuuji Ichisugi[1]　Hidemoto Nakada[1]　Naoto Takahashi[1]　Takashi Sano[2]

[1] 産業技術総合研究所 人工知能研究センター

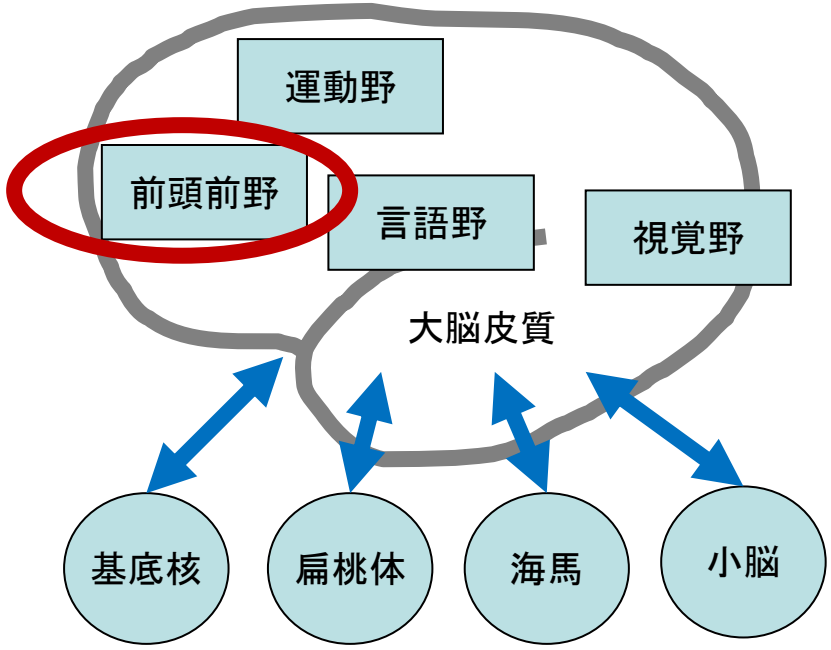
[1] National Institute of Advanced Industrial Science and Technology (AIST), AIRC

[2] 成蹊大学 理工学部 情報科学科

[2] Department of Computer and Information Science, Faculty of Science and Technology, Seikei University

私の研究の中期的目標

- 前頭前野周辺の計算論的モデルの構築
  - 報酬期待値を最大化する合理的な行動・思考・言語理解・発話をする知的エージェントのモデル



背景と発表の概要

- 記号AIと統計的機械学習の統合は重要な未解決問題
- 我々は以前 Prolog 言語が行うような推論を階層型強化学習 RGoal の上で実現する手法を提案

  [一杉他 第12回 汎用人工知能研究会 2019年8月]

  – ただし学習は未実装、正しい推論規則をあらかじめ与えていた
- 今回は推論規則を経験から獲得する方法を提案

  – その一部である推論規則の「正しさ」の学習方法を実装

今回の目標：
推論の正しさの学習

「きのう戸棚にチョコレートがあった。 ということは、きょうもあるはず。」

「あれ？チョコレートがない。
そういえば、おにいちゃんが今朝食べたと言っていた。
食べたからなくなったのか・・・。」

「きのうチョコレートがあったらきょうもチョコレートがある」とは
限らない、と学習。

https://www.irasutoya.com/2016/09/blog-post_280.html

https://www.irasutoya.com/2016/12/blog-post_40.html

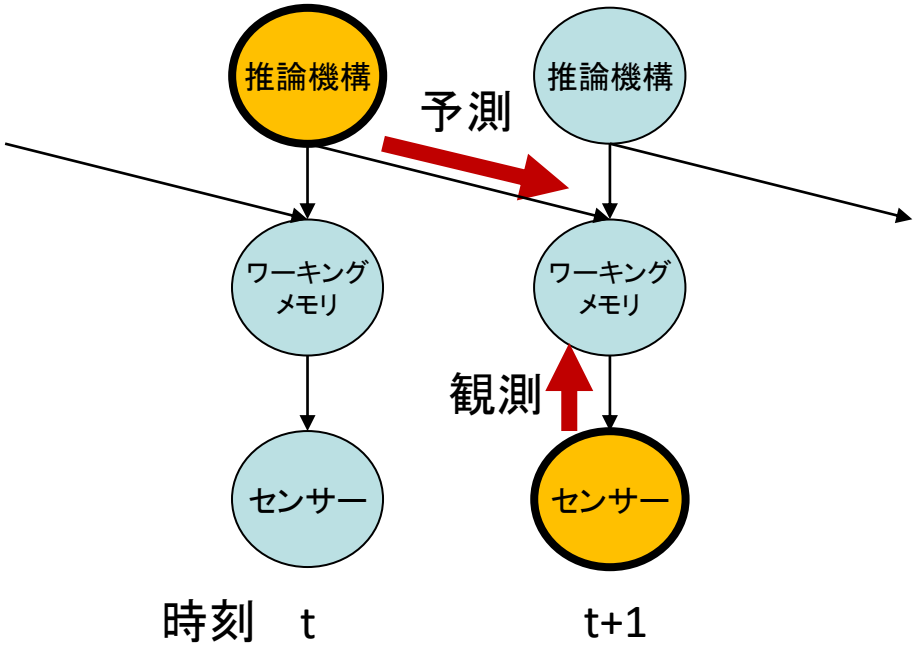
記号AIと強化学習の融合の試み

- RRL (Relational Reinforcement Learning, 2001)
  - **高い汎化能力の実現が目的**
    - 強化学習と帰納論理プログラミングの融合
    - 行動価値関数を論理決定木に圧縮
- Soar-RL (2005)
    - プロダクションシステムのルールの優先度の学習
- rlCoP (2018) NeurIPS
  - **定理証明システムの性能向上が目的**
    - 証明の1ステップをモンテカルロ木探索で選択
- 本研究
  - **ヒトと同様の推論・知識獲得の再現が目的**
  - **生物学的妥当性とマルチタスク環境での効率的学習を重視**

エージェントのアーキテクチャ

ワーキングメモリの値は、環境において成り立っているとエージェントが信じる命題の集合

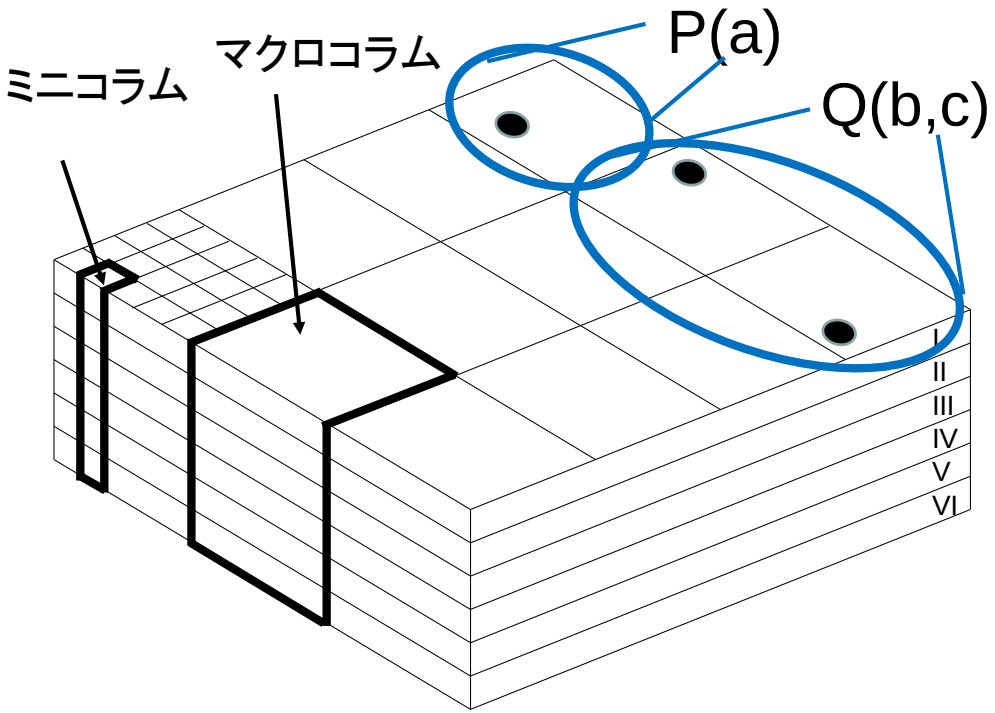
例：「戸棚の中にチョコレートがある」「外は晴れている」・・・



推論機構はワーキングメモリの現在の状態から次の状態を予測・推論

大脳皮質と述語の集合

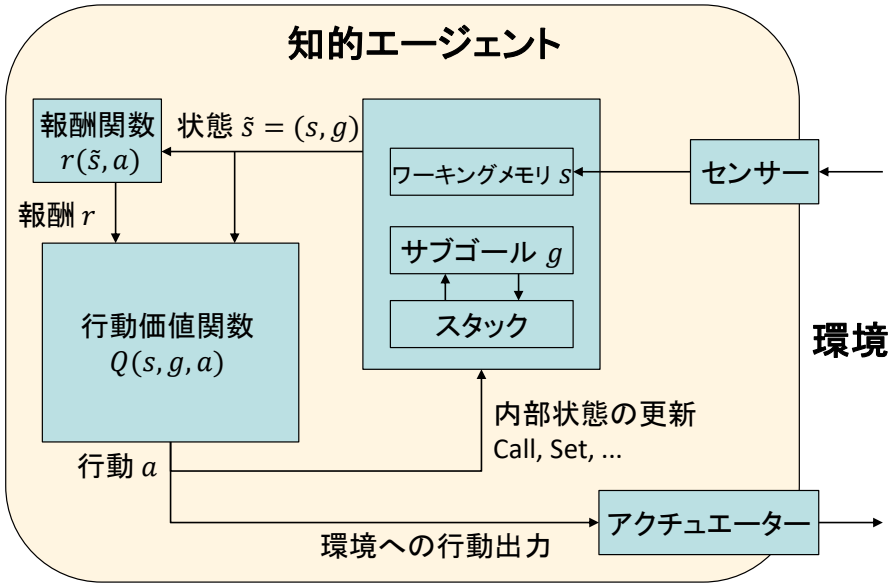
大脳皮質	ベイジアンネット	述語
マクロコラム	多値確率変数	述語の引数
ミニコラム	確率変数の値	述語の引数の値
ニューロン発火	事後確率	真偽値
コンフリクト検出	同時確率が0	矛盾



階層型強化学習 RGoal

[一杉 et al.第9回 汎用人工知能研究会 2018]

- 再帰的なサブルーチン呼び出しが可能。
  - サブルーチン＝サブゴールに向かう方策
- マルチタスク環境で学習を加速。
- サブルーチンを組み合わせて未経験のタスクも解ける。
  - 有限個の推論規則の組み合わせで無限の推論タスクが解ける。



パターンを用いたテーブルの圧縮

[一杉 et al.第10回 汎用人工知能研究会 2018]

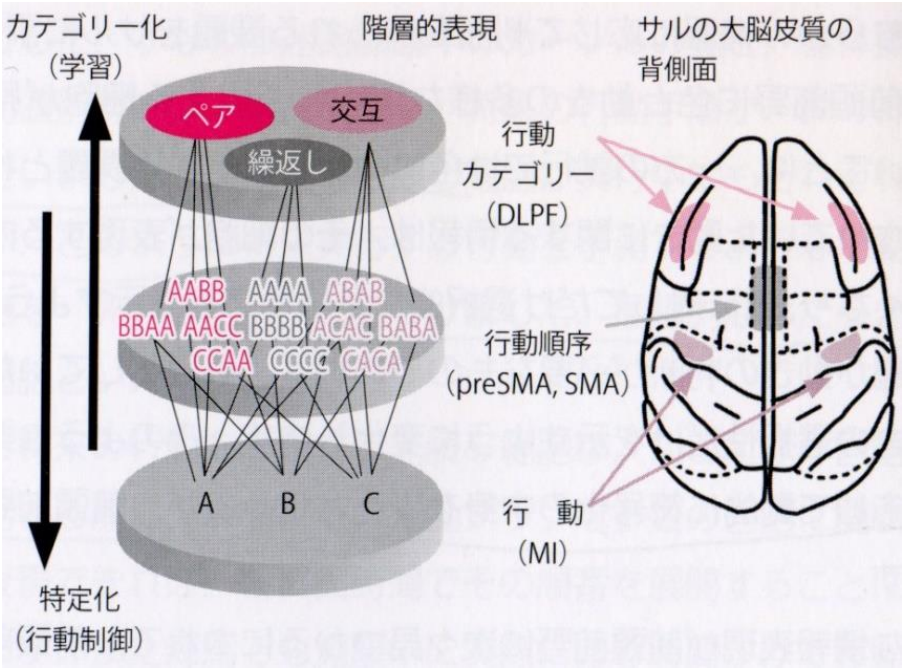
X \ Y	0	1	2	3	4
0	2.0	1.0	1.0	3.0	1.0
1	1.0	2.0	1.0	3.0	1.0
2	1.0	1.0	2.0	3.0	1.0
3	1.0	1.0	1.0	4.0	1.0
4	1.0	1.0	1.0	3.0	2.0

パターン	値
(3,3)	4.0
(X,3)	3.0
(X,X)	2.0
(X,Y)	1.0

テーブルのインデックスの代わりにパターンを用いることで、
サイズ 5 x 5 = 25 のテーブルがサイズ 4 に圧縮

・ルールの順序には意味はないものとする。
・マッチするルールが複数ある場合、最も特殊なパターンをもつルールの方を選択する。
・「最も特殊なパターン」とは、用いる変数の種類が最も少ないパターンと定義。

前頭葉とパターン



前頭前野外側部で動作順序をパターンで抽象化した動作カテゴリとして表現

Keisetsu Shima, Masaki Isoda, Hajime Mushiake and Jun Tanji,
Categorization of behavioural sequences in the prefrontal cortex,
Nature, 445, 315-318, 2007.

図：虫明 元「前頭葉のしくみ―からだ・心・社会をつなぐネットワーク― 」
共立出版, 2019.

推論規則を経験から獲得する際の2つの問題と解決策

- **探索空間が広すぎる。**
  →「正しい推論規則の候補の獲得」と、
  「推論規則の価値の学習」の2段階に分けて知識を獲得するというシナリオを提案

- **正解がわからないと推論規則が正しいかどうかも判断できない。**
  推論の目的の1つは隠れた状態の推定
  例:「戸棚の中にチョコレートはあるか？」

  →「通常モード」と「推論訓練モード」を使う学習機構を提案
  今回、推論訓練モードのモデルを実装し動作を確認

「正しい推論規則の候補」の獲得

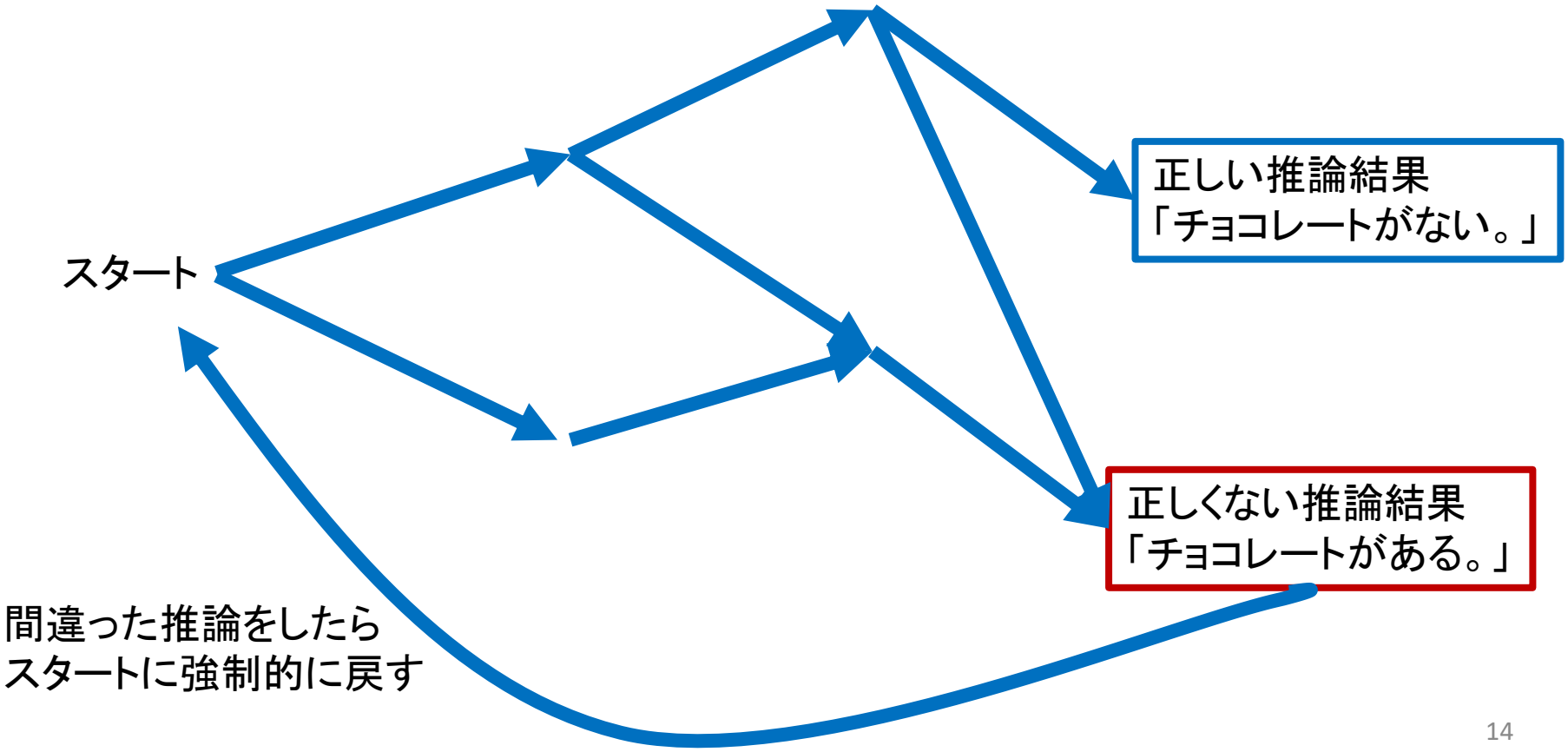
- **人間は1つの経験からワンショットで推論規則の候補を獲得する。（ただし、その推論規則が常に正しいかどうかはわからない。）**
  - 「カラスを見た。そばにカメがいた。」という経験から
    「カラスがいたらそばにカメもいる。」という推論規則を獲得
  - 「隣の家の太郎君が犬をかわいがっていたよ。」という発話から
    「太郎君は犬が好きである。」
    「隣の家の人はみんな犬が好きである。」
    「太郎君は動物が好きである。」
    という推論規則を獲得
  - 目立つ特徴・事象に注目、さらに経験した事実を抽象化
  - この機構は今回は未実装
- **こうやって獲得した「正しい推論規則の候補」の「正しさ」を、次に説明する推論訓練モードで学習**

学習の「正解」を得る方法：通常モードと推論訓練モード

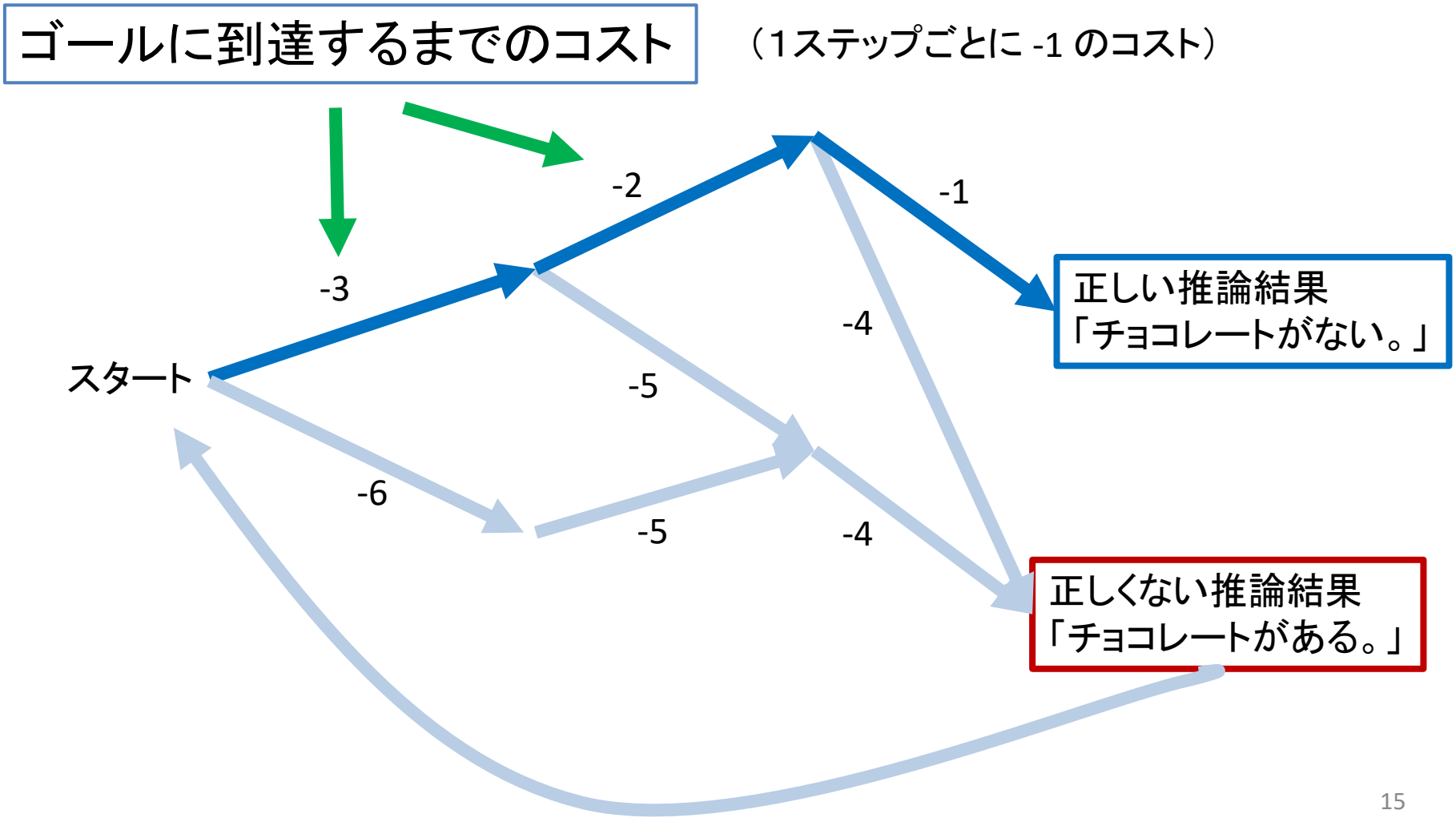
- 通常モード：
  - 通常の強化学習と同様に行動・学習するモード。
  - 予測と観測事実との矛盾の検出で推論訓練モードに移行。
    例：「チョコレートがあると思ったのに、ない！」
- 推論訓練モード（**今回実装**）：
  - 推論の「正解」をゴールにして推論（証明探索）を繰り返すモード。
  - 「正解」はたった今、予想に反して観測したその命題そのもの用いる。
    例：「チョコレートがない。」をゴールにして推論を繰り返す。
  - 推論結果が「正解」と異なる場合は推論を最初から繰り返す。
    例：「チョコレートがあるはず。」→「やっぱりない。なんでだろう。」
  - 何度も繰り返すと、**今の状況で「チョコレートがあるはず」という結果を導くことに関与する状態行動対の価値が下がっていき**、やがて「チョコレートがない」という正しい推論をするようになるはず。

推論の「正しさ」を「正解に到達するまでのコストの低さ」で代用

推論＝証明探索＝ゴールに向けた経路探索



強化学習によりゴールへの最短経路が学習される



行動ルール

- 行動ルール: rule(s,g,a)
  - s :現在のワーキングメモリの値
  - g :サブゴール
  - a :行動
    - Call(g') はサブルーチン呼び出し: 現在のサブゴール g をスタックに積んで新たなサブゴールを g' に設定
    - Set(s') はワーキングメモリの値を s' と環境の状態をパターンマッチした結果の値に設定
      ただし環境の状態と矛盾したら fail
    - Fail はスタックの状態を破棄しエージェントを初期状態にリセット

ワーキングメモリは環境にグラウンディング

- 1つの行動ルールは背外側前頭前野のミニコラム1つに対応すると想定(ヒトで100万個程度)

実行が fail したときの学習則

拡張状態行動空間 [Ichisugi et.al 2019] 上での Sarsa

$$Q_G((s,g),a) \leftarrow Q_G((s,g),a)$$
$$+\alpha(r + Q_G((s',g'),a') - Q_G((s,g),a)) \quad (1)$$

実行が fail したときの学習則

$$Q(s,g,a) \leftarrow Q(s,g,a)$$
$$+\alpha(r + Q(s',g',a') - Q(s,g,a) - V(g,stack)) \quad (4)$$

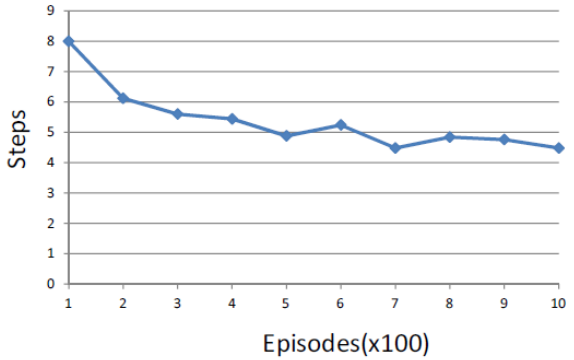
ただし

$$V(g,stack) \equiv V_{g_1}(g) + V_{g_2}(g_1) + \cdots + V_{g_n}(g_{n-1})$$
$$V_g(s) \equiv \Sigma_a \pi((s,g),a) Q(s,g,a) \quad (3)$$

サンプル実装：https://staff.aist.go.jp/y-ichisugi/besom/InfLearn20200220.zip

実行例1：正しくない推論規則の価値が下がる例

環境において
「P(1) と Q(2) が成り立っている」か
「P(2) と Q(1) が成り立っている」かの
いずれかのとき、
命題 Q(a) を推論するタスクで学習



エピソード終了までの平均実行項時間

あらかじめエージェントに与える行動ルール集合

1:	rule(ANY, Q(_), Call(P(_)))	-3.2281303
2:	rule(P(1), Q(_), Set(Q(2)))	-0.9949911
3:	rule(P(2), Q(_), Set(Q(1)))	-0.99138117
4:	rule(ANY, P(_), Set(P(_)))	-0.999997
5:	*rule(P(1), Q(_), Set(Q(1)))	-3.1717572
6:	*rule(P(2), Q(_), Set(Q(2)))	-2.9107559

P(1) ならば Q(2)

P(2) ならば Q(1)

P(1) ならば Q(1)

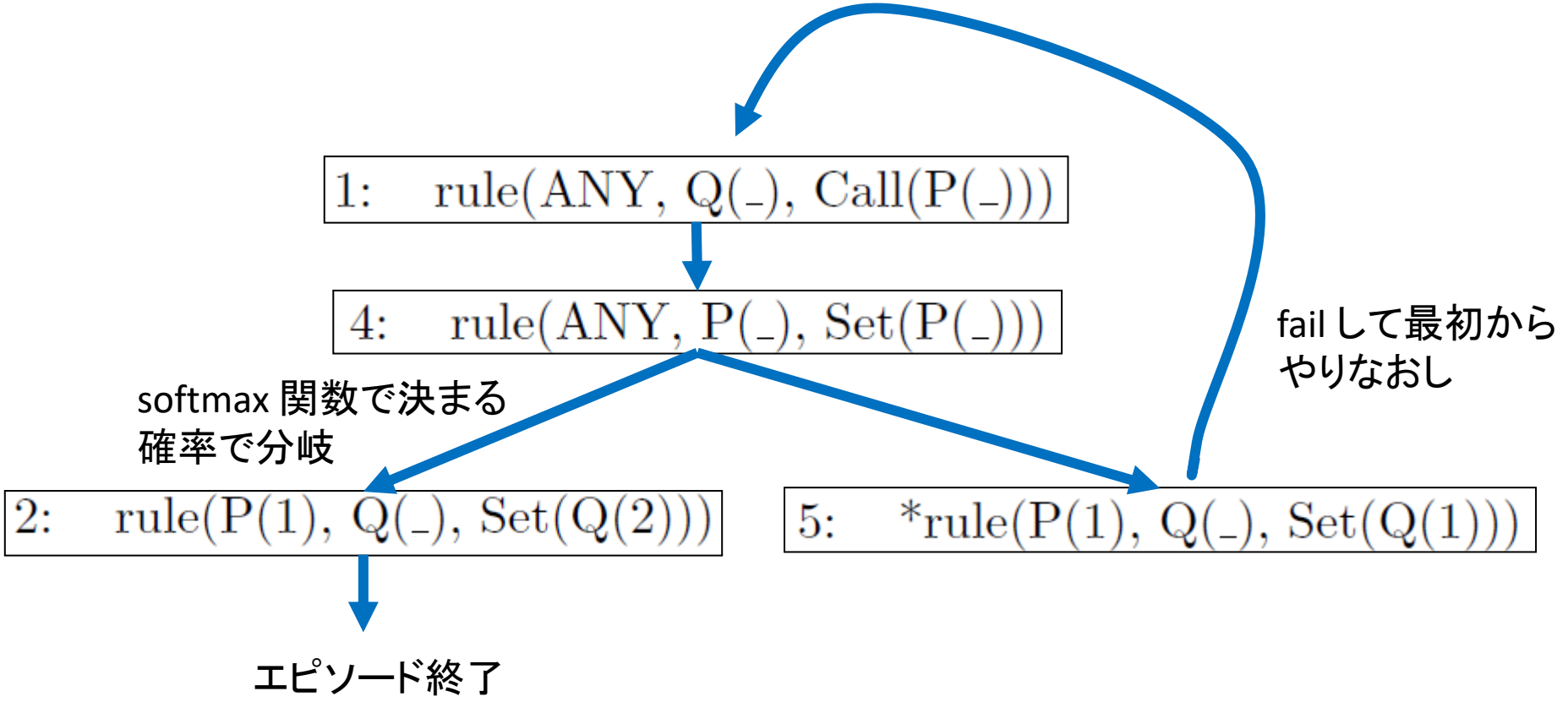
P(2) ならば Q(2)

正しくないので
価値が下がる

1000エピソード学習後の価値

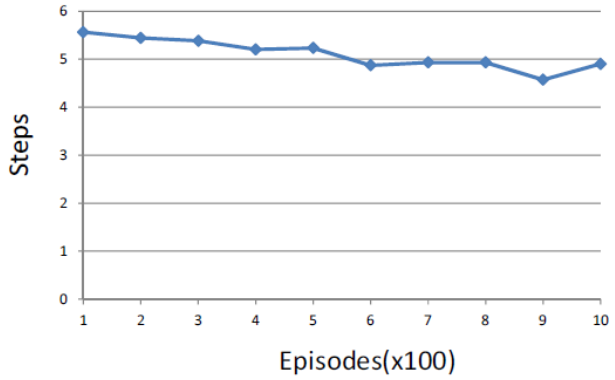
実行の流れ

環境の状態 : { P(1), Q(2) }



実行例2：遠回りの推論が使われなくなる例

環境において
「P(1) と Q(1) と R(1) が成り立っている」か
「P(2) と Q(2) と R(2) が成り立っている」かの
いずれかのとき、
R(a) を推論するタスクで学習



エピソード終了までの平均実行項時間

1:	rule(ANY, R(_), Call(P(_)))	-2.9817963
2:	rule(P(X), R(_), Set(R(X)))	-0.9982739
3:	rule(ANY, R(_), Call(Q(_)))	-4.2641506
4:	rule(Q(X), R(_), Set(R(X)))	-0.9749899
5:	rule(ANY, Q(_), Call(P(_)))	-2.80906
6:	rule(P(X), Q(_), Set(Q(X)))	-0.9749899
7:	rule(ANY, P(_), Set(P(_)))	-0.9999568

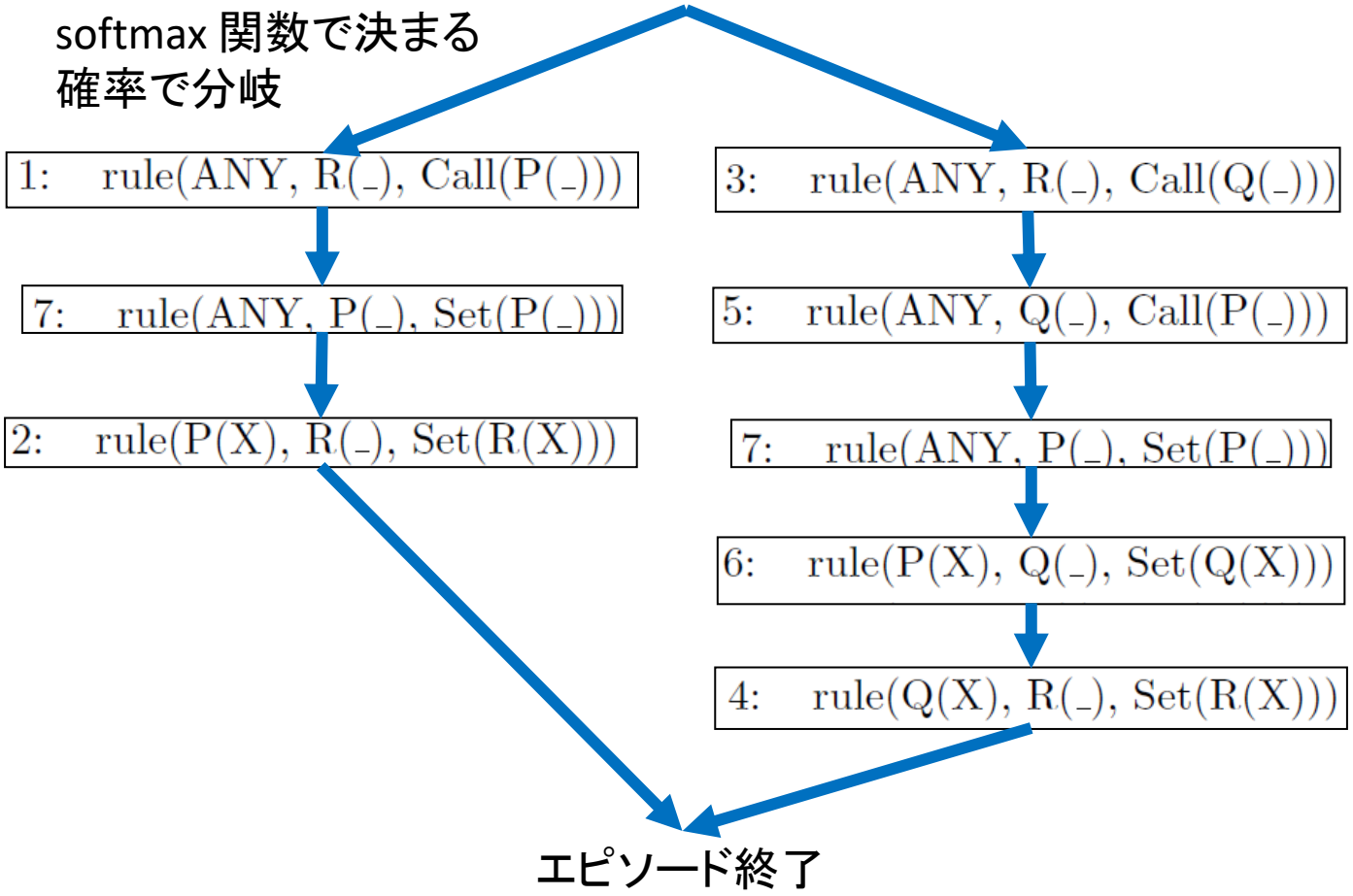
1000エピソード学習後の価値

P(X) から R(X) を推論

P(X) から Q(X) を推論してから
Q(X) から R(X) を推論

遠回りなので価値が下がる

実行の流れ


softmax 関数で決まる
確率で分岐
1: rule(ANY, R(_), Call(P(_)))
7: rule(ANY, P(_), Set(P(_)))
2: rule(P(X), R(_), Set(R(X)))
3: rule(ANY, R(_), Call(Q(_)))
5: rule(ANY, Q(_), Call(P(_)))
7: rule(ANY, P(_), Set(P(_)))
6: rule(P(X), Q(_), Set(Q(X)))
4: rule(Q(X), R(_), Set(R(X)))
エピソード終了

そのほかの実行例

実行例3：間違ったサブルーチンを呼び出す行動ルールが使われなくなる例

1:	rule(ANY, R(_), Call(P(_)))	-2.9989986
2:	rule(P(X), R(_), Set(R(X)))	-0.9999568
3:	rule(ANY, R(_), Call(Q(_)))	-5.0751033
4:	rule(Q(X), R(_), Set(R(X)))	-3.8932161
5:	rule(ANY, P(_), Set(P(_)))	-0.9999568
6:	*rule(ANY, Q(_), Set(Q(2)))	-0.98258275

実行例4：決めつけの方が価値が高い例

1:	rule(ANY, Q(_), Call(P(_)))	-2.7919052
2:	rule(P(1), Q(_), Set(Q(2)))	-0.8824299
3:	rule(P(2), Q(_), Set(Q(1)))	-0.9446731
4:	rule(ANY, P(_), Set(P(_)))	-0.9934953
5:	*rule(ANY, Q(_), Set(Q(1)))	-1.987416

実行例5：2つの命題から値を推論する例

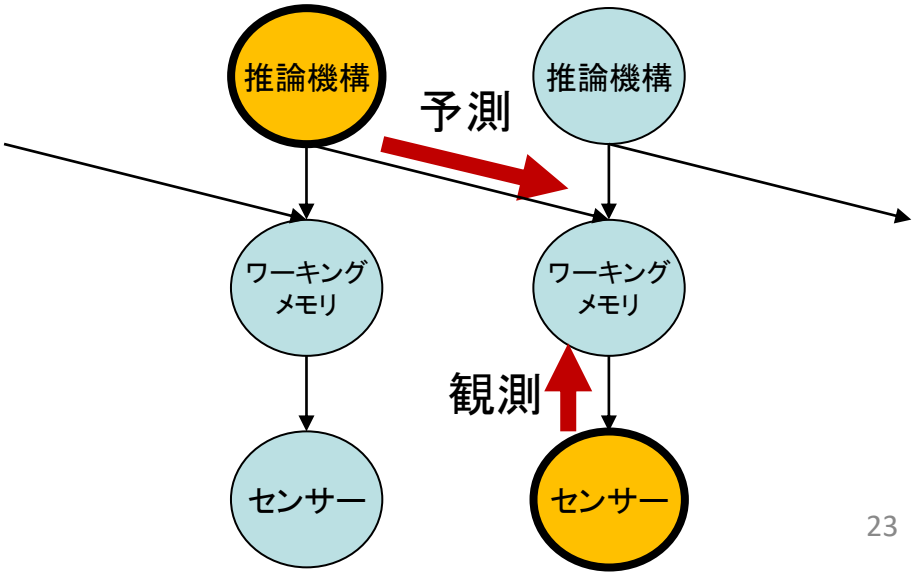
1:	rule(ANY, R(_), Call(P(_)))	-6.830637
2:	rule(P(X), R(_), Call(AND(P(X),Q(_))))	-4.8854413
3:	rule(P(X), AND(P(X),Q(_)), Call(Q(_)))	-2.9989986
4:	*rule(Q(Y), AND(P(X),Q(_)), Set(AND(P(X),Q(Y))))	-0.9999568
5:	rule(AND(P(1),Q(1)), R(_), Set(R(1)))	-0.90480024
6:	rule(AND(P(1),Q(2)), R(_), Set(R(2)))	-0.91729563
7:	rule(AND(P(2),Q(1)), R(_), Set(R(2)))	-0.935021
8:	rule(AND(P(2),Q(2)), R(_), Set(R(1)))	-0.9156164
9:	rule(ANY, P(_), Set(P(_)))	-0.9999568
10:	rule(ANY, Q(_), Set(Q(_)))	-0.9999568

実行例6：例外的な状況を処理するルールの例

1:	rule(ANY, Q(_), Call(P(_)))	-2.8419564
2:	*rule(P(X), Q(_), Set(Q(X)))	-0.9996545
3:	rule(P(5), Q(_), Set(Q(10)))	-0.8751221
4:	rule(ANY, P(_), Set(P(_)))	-0.9999568

予測と観測の不一致の検出機構

- 提案手法では2か所で使っている
  - 通常モードから推論学習モードへの切り替えタイミングの検出
  - 推論訓練モードにおける fail すべき状況の検出
- 大脳皮質全体がベイジアンネットならば、同時確率が0であることを検出することで実現可能

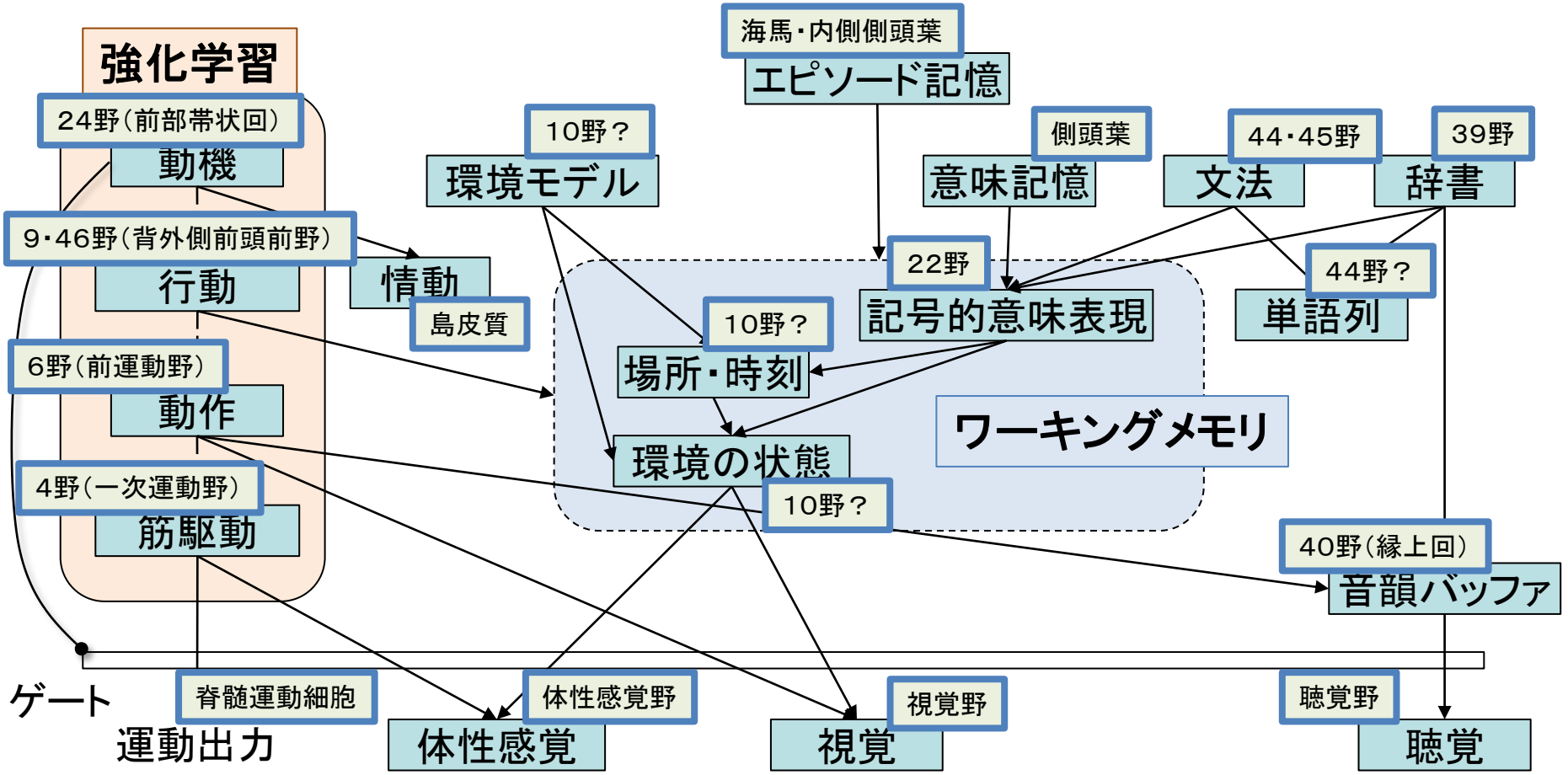


まとめと今後

- 環境の中で動く知的エージェントが正しい推論規則を経験から獲得するための手法を提案
- 提案手法は、推論規則の「正しさ」を「正解に到達するまでのコストの低さ」で代用し、それを階層型強化学習 RGoal によって学習
- 予測と矛盾する観測結果を「正解」として推論訓練モードを実行
- 汎用人工知能の自律的な知識獲得の基本原理の1つの候補

- 今後:
  - 他者の心の状態の推定を伴う言語理解、合目的的な発話計画などのモデルの実現、対話システムへの応用
  - 脳全体のアーキテクチャの実現

脳型AGIアーキテクチャ全体像（案）



脳全体のアーキテクチャの実現に向けて実装すべきこと

- 注意の機構を用いた「正しい推論規則の候補」の獲得
- 通常モードと推論訓練モードの切り替え機構
- 異なる時刻・場所における複数のイベントの表現
- 一時メモリと永続メモリ(Cf. Dyna-2)
- 宣言的知識（エピソード記憶と意味記憶）
- メタ認知・メタ学習

記号AIが扱う知識の利点

- **知識の汎用性が高い。**
  - 知識を組み合わせることで多様な推論が行える。
- **汎化能力が高い。**
- **知識のモジュラリティが高い。**
  - 環境の変化に対応しやすい。
- **自然言語との相互変換が容易。**
  - 他者との情報交換を前提とした知識の内部表現に適している。