

講義 2: 演繹推論エージェント
Deductive Reasoning Agents

内容

1. エージェントアーキテクチャ
2. 記号推論エージェント
 - (a) 演繹推論エージェント
3. AGENT0

1. 演繹推論とは?(1/3)

- 演繹推論 (deductive reasoning)
- 帰納推論 (inductive reasoning)
- 仮説生成 (abductive reasoning) (cf. 仮説推論)

1. 演繹推論とは?(2/3)

- 演繹推論

形式的体系において、その公理およびその他のいくつかの論理式から出発して推論規則を繰り返し適用することにより1つの論理式が導き出される過程を形式的に表現したもの。例

- 帰納推論

与えられた個々の事実から一般的な規則を導き出す推論をいい、一般的な規則から個別の事実を説明する。

- 仮説生成

与えられた結論あるいは観測事象を論理的に説明できるように適当な仮説を探し出して、公理系や知識を補う推論方式を言う。

2. エージェントアーキテクチャ (1/3)

- 柔軟な自律行為が行えるコンピュータシステムとしてのエージェントの考え方の導入.
- エージェントに基づくシステムを構築するための研究課題の簡単な紹介.
- 3タイプのエージェントアーキテクチャ
 - 記号的/論理的/熟考的
 - 即応的
 - ハイブリッド

2. エージェントアーキテクチャ (2/3)

- 自律性, 即応性, 熟考性, 社会性などの特性を兼ね備えたエージェントの構築が目的.
- Patti Maes によるエージェントアーキテクチャの定義
“[A] particular methodology for building [agents]. It specifies how... the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact. The total set of modules and their interactions has to provide an answer to the question of how the sensor data and the current internal state of the agent determine the actions... and future internal state of the agent. An architecture encompasses techniques and algorithms that support this methodology.”

3. エージェントアーキテクチャ (3/3)

- 従来 (1956-1985), AI で設計されたほとんどすべてのエージェントは**記号推論**エージェントであった。エージェントが何をするのかを決定するのに**明示的**な論理推論を用いる。
- 記号推論に付随する問題に対し**即応**というアイデアが生まれ (Brooks), 1985 年から現在に至るまでそれは**即応エージェント**ムーブメントと呼ばれている。(refer: Subsumption Architecture, 包摂アーキテクチャ)
- 1990 年から現在まで, 多くの**ハイブリッド**アーキテクチャ (推論と即応の組合せ) が提案されている。(refer: メタレベルアーキテクチャ)

3. 記号推論エージェント (1/3)

- エージェント構築の古典的アプローチは、エージェントを**知識ベースシステム**とみなし、そのシステムに関連した方法論を寄せ集めたもの。
- このパラダイムは**記号的 AI**と呼ばれる。
- 熟考エージェントを次のように定義する。
 - 世界に関する明示的な記号モデルを含む；
 - 記号推論により意志決定（例えばどの行為を実行するか）を行う。

3. 記号推論エージェント (2/3)

このアプローチでエージェントを構築する場合、解決すべき二つの重要な問題がある。

1. **変換問題** (transduction problem) 実世界の問題を、有用性を損なわないうちに正確で適切な世界の記号的記述に変換する問題。
2. **表現/推論問題** (representation/reasoning problem) 有用性を損なわないうちに、情報を記号的に表現し、エージェントにそれを操作/推論させる問題。
 - 前者の問題は視覚，音声理解，学習などの研究となった。
 - 後者の問題は知識表現，推論，プランニングなどの研究を生じさせた。

3. 記号推論エージェント (3/3)

これらの問題が生み出したおびただしい量の研究にもかかわらず、

- いずれの問題も解決に近付いていないと多くの人が主張する。
- 根本的な問題は記号操作アルゴリズムの複雑さに起因する。
- 常識推論のような一見したところ自明な問題であっても、非常に難しいことが判明した。

⇒ これらの問題によりエージェント構築の他のアプローチが探索された
(再来週以降の講義話題)。

3.1 演繹推論エージェント (1/6)

定理証明器としてのエージェントのアプローチ.

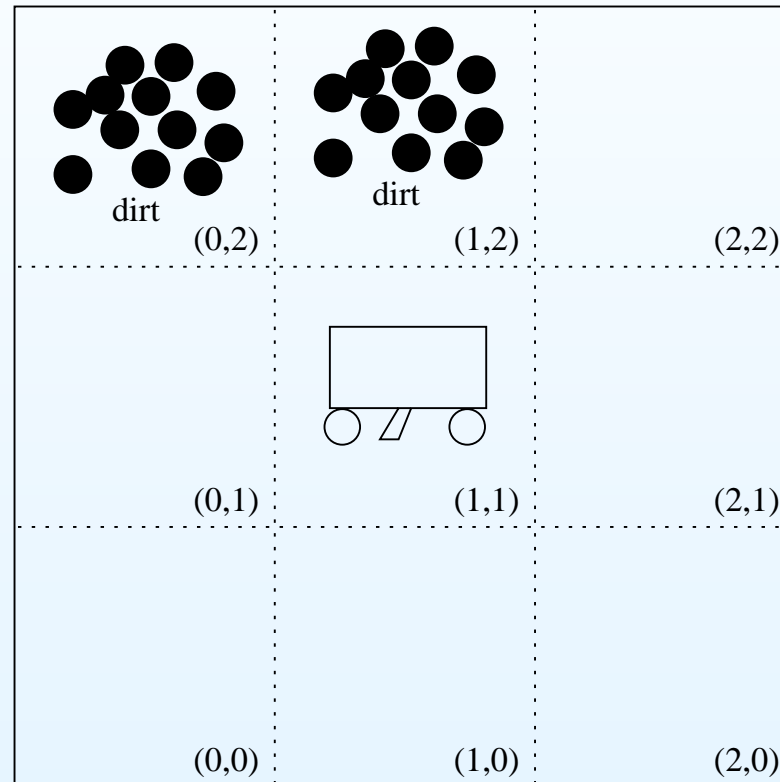
- エージェントは, 定理証明を用いどのようにして, 何をするかを決定できるのか?
- 基本的な考え方: 任意の与えられた状況における**最良の行為**を表す理論を記号化するのに論理を用いる.
- 記法の導入.
 - ρ : その理論 (典型的には演繹規則の集合)
 - Δ : 世界の現在の状態を記述する論理データベース
 - Ac : エージェントが実行できる行為の集合
 - $\Delta \vdash_{\rho} \varphi$: 論理式 φ が演繹規則 ρ だけを用いてデータベース Δ から証明される

3.1 演繹推論エージェント (2/6)

```
Function: Action Selection as Theorem Proving
1.  function action( $\Delta$ ) returns an action  $A_c$ 
2.  begin
      /* try to find an action explicitly prescribed */
3.    for each  $\alpha \in A_c$  do
4.      if  $\Delta \vdash_{\rho} Do(\alpha)$  then
5.        return  $\alpha$ 
6.      end-if
7.    end-for
      /* try to find an action not excluded */
8.    for each  $\alpha \in A_c$  do
9.      if  $\Delta \not\vdash_{\rho} \neg Do(\alpha)$  then
10.       return  $\alpha$ 
11.     end-if
12.   end-for
13.   return null /* no action found */
14. end function action
```

3.1 演繹推論エージェント (3/6)

例題: 掃除機の世界 (Vaccume World).
ロボットの目標はすべてのゴミを掃除すること!



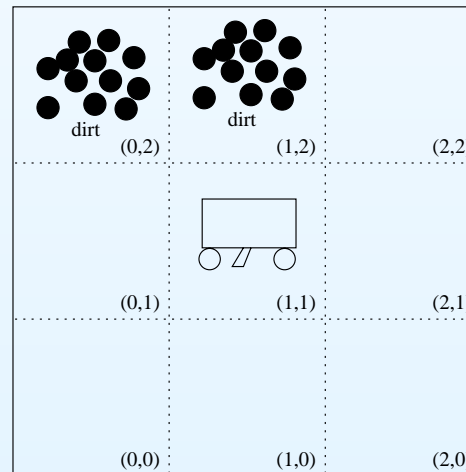
3.1 演繹推論エージェント (4/6)

- この例題では以下の3つの領域述語を用いる。
 - $In(x, y)$: エージェントが (x, y) にいる
 - $Dirt(x, y)$: (x, y) にゴミがある
 - $Facing(d)$: エージェントが方向 d を向いている。
- 可能な行為。
 $Ac = \{turn, forward, suck\}$

(注: *turn* は右折を意味する)

3.1 演繹推論エージェント (5/6)

- 何をすべきかの規則 ρ .
 - $In(x, y) \wedge Dirt(x, y) \longrightarrow Do(suck)$ (基本的な掃除行為)
 - $In(0, 0) \wedge Facing(north) \wedge \neg Dirt(0, 0) \longrightarrow Do(forward)$
 - $In(0, 1) \wedge Facing(north) \wedge \neg Dirt(0, 1) \longrightarrow Do(forward)$
 - $In(0, 2) \wedge Facing(north) \wedge \neg Dirt(0, 2) \longrightarrow Do(turn)$
 - $In(0, 2) \wedge Facing(east) \longrightarrow Do(forward)$



- これらの規則 (と他の明白な規則) を使って (0, 0) から出発すればロボットはゴミを掃除できる。

3.1 演繹推論エージェント (6/6)

- **問題点**
 - ビデオカメラからの入力をどのようにして $Dirt(0, 1)$ に変換するか?
 - 意志決定は静的環境を仮定している (計算合理性)
 - 一階述語論理を用いた意志決定は**決定不能問題**.
- 仮に命題論理を用いたとしても, 最悪ケースでの意志決定は co-NP 完全問題と同じだけの計算量が必要 (これは悪い知らせ!).
- **典型的な解法**
 - 論理を弱める
 - 記号的であるが非論理的な表現を用いる
 - 推論の強調を実行時から設計時に移す

4. AGENT0(1/10)

- AI 研究コミュニティのエージェントに対する興味の多くは Shoham のエージェント指向プログラミング (AOP) の概念から生じてきた。
- 新計算パラダイム AOP は計算の社会的考察に基づく。
- AOP の重要なアイデアは、エージェントの特性を表現するために心的概念 (信念, 願望, 意図など) により, エージェントを直接プログラミングすること。
- この提案の背後にある動機は, 人間は複雑なシステムの特性を表現する抽象化メカニズムとして, そのような概念を使うということ。

⇒ 人間の振舞いの記述や説明に心的概念を使うのと同じように, 機械をプログラムするのにそれらを使用したら有用であろう!! でも, なぜ?

4. AGENT0(2/10)

- 完全な AOP システムは 3 つの構成要素を持つ [Shoham].
 - エージェントの心的状態を規定し記述する論理.
 - エージェントをプログラミングするインタプリタ型プログラミング言語.
 - 自然なアプリケーションをエージェントに変換するための**エージェント化** (agentification) プロセス.

研究論文では最初の二つの構成要素について報告している.

- ここでは論理をスキップして AOP 言語について考える.

4. AGENT0(3/10)

- AGENT0は LISP の拡張として実装された。AGENT0 のエージェントは 4 つの構成要素を持つ。
 - 能力 (エージェントが出来ること) の集合
 - 初期信念の集合
 - 初期コミットメントの集合
 - コミットメント規則の集合
- 鍵となる構成要素はコミットメント規則の集合

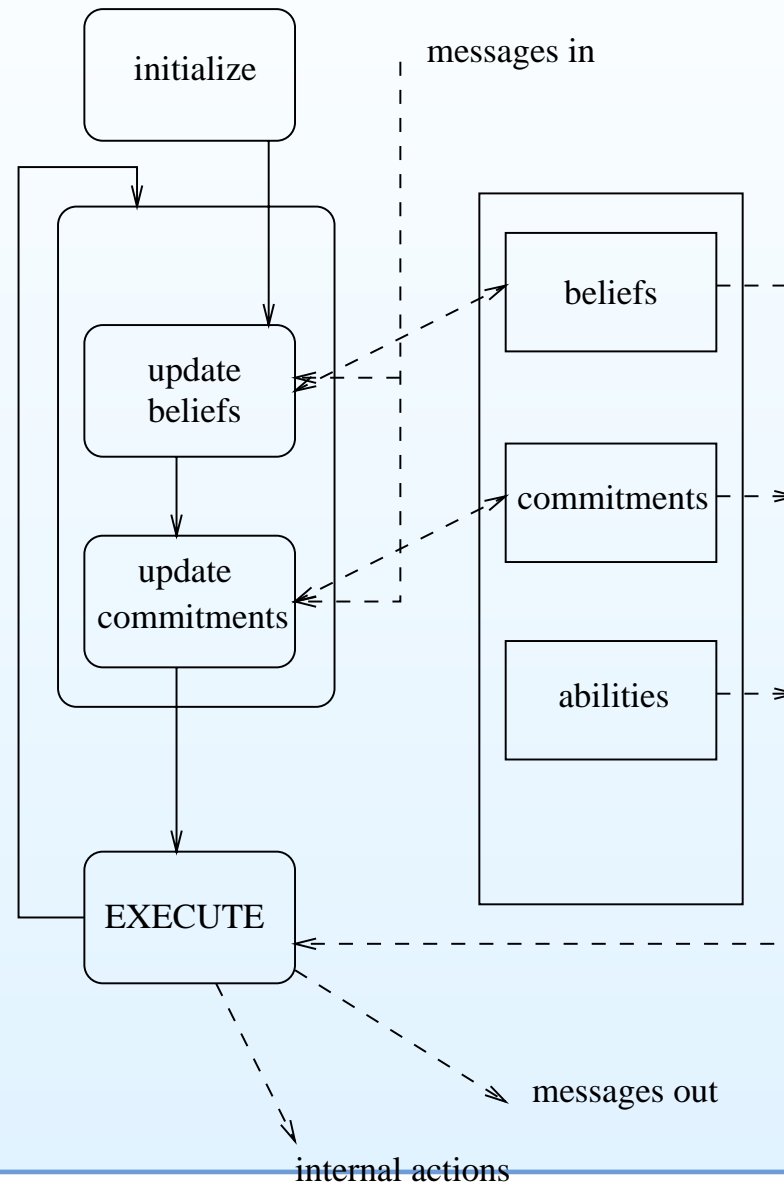
4. AGENT0(4/10)

- 各コミットメント規則は以下を含む。
 - メッセージ条件
 - 心的条件
 - 行為
- 各エージェントサイクルで、規則が発火するかどうかを決定するために、
 1. エージェントが受信したメッセージに対してメッセージ条件が照合される
 2. エージェントの信念に対して心的条件が照合される
 3. もしも規則が発火すると、エージェントはその行為にコミットする

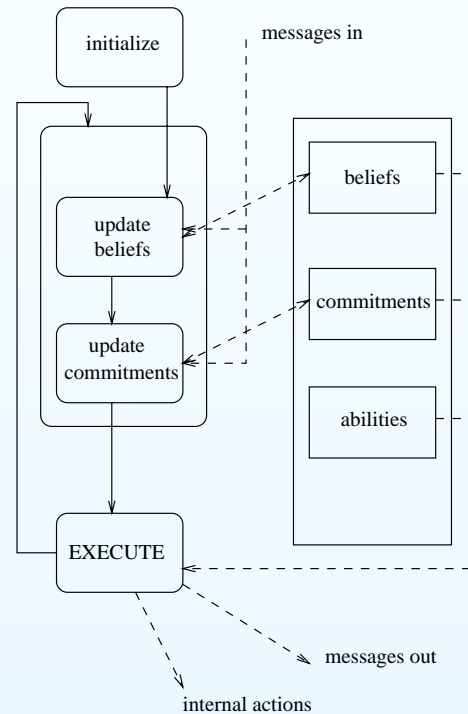
4. AGENT0(5/10)

- 行為は以下の 2 種類
 - 私的 (private): 内部実行されるサブルーチンに対応するもの
 - 発話的 (communicative): メッセージ送信
- メッセージは以下の 3 つのタイプのうちの一つに制限される
 1. 要求 (request): 行為へのコミットメントを要求する
 2. 非要求 (unrequest): 行為を行わないよう非要求する
 3. 伝達 (inform): 情報を伝える

4. AGENT0(6/10)



4. AGENT0(7/10)



1. すべての現在のメッセージを読み，必要な場合には信念，そしてそれによりコミットメントを，更新する。
2. 関連づけられた行為の能力条件が満たされたら，現在のサイクルに対するすべてのコミットメントを実行する。
3. (1) へ行く。

4. AGENT0(8/10)

- コミットメント規則の例.

```
COMMIT(  
  ( agent, REQUEST, DO(time, action)  
  ), ;;; msg condition  
  ( B,  
    [now, Friend agent] AND  
    CAN(self, action) AND  
    NOT [time, CMT(self, anyaction)]  
  ), ;;; mental condition  
  self,  
  DO(time, action)  )
```


4. AGENT0(9/10)

- この規則は次のように言い替えられる

もし私が *time* において *action* を行うことを要求するメッセージを *agent* から受信したら、私は以下のことを信じ、

- *agent* は現在、友人である;
- 私はその行為を実行できる;
- *time* において、如何なる行為にもコミットしていない、

そして、*time* において *action* を行うことにコミットする。

4. AGENT0(10/10)

- AGENT0はマルチエージェントにおける協調と対話を支援する。
- しかし，AGENT0はある原理を例証するためのプロトタイプであり，生産言語ではない。
- より洗練された PLACA という実装がある [Thomas1993]

1. 演繹推論とは?(1/3)

- 埃が目に入る → 盲人が増える
- 盲人が増える → 三味線屋が儲かる
- 風が吹く → 埃が立つ
- 鼠が増える → 桶がたくさんかじられる
- 三味線屋が儲かる → 猫が減る
- 猫が減る → 鼠が増える
- 埃が立つ → 埃が目に入る
- 桶がたくさんかじられる → 桶が減る
- 桶が減る → 桶屋が儲かる

戻る