第 9 章

知的エージェント

「知能コンピューティング論 テキスト (阪大　馬場口 登©)」　(Dec/2002)

近年，**「知的エージェント (intelligent agent)」**あるいは**「エージェント」**という術語が，人工知能分野のキーワードになりつつある [1,2]．特に，インターネットのような大規模ネットワーク環境において，地理的に分散し機能的に独立したプログラムを統合利用する場合や，知能ロボットを未知の環境で動作させる場合に，知的エージェントという考え方が重要になると予想されている [3]．さらに，応用レベルの技術として，**「ソフトウェアエージェント」**，**「エージェント指向インタフェース」**が，そしてインプリメンテーションレベルの技術として，**「エージェント指向プログラミング」**，**「モバイルエージェント」**が，また，分散協調処理系として，**「マルチエージェント」**も注目されている．本章では，知的エージェントの基礎について述べる．但し，未だ体系化されるには至っておらず，決着のついていない議論も多いことを最初に指摘しておく．

9.1　知的エージェントのモデル

図 9.1 は知的エージェントのモデル図である [1,2]．知的エージェントは，その外の環境とインタラクションを取りながら，何らかのタスクを実行するものである．

いま，エージェントが対峙する外部環境を E と記す．エージェントの内部は，大きく 3 つの部分，すなわち**知覚部** (perceiver)，**推論部** (reasoner)，**作用部** (effector) からなる．それぞれは以下の働きを担う部分である．

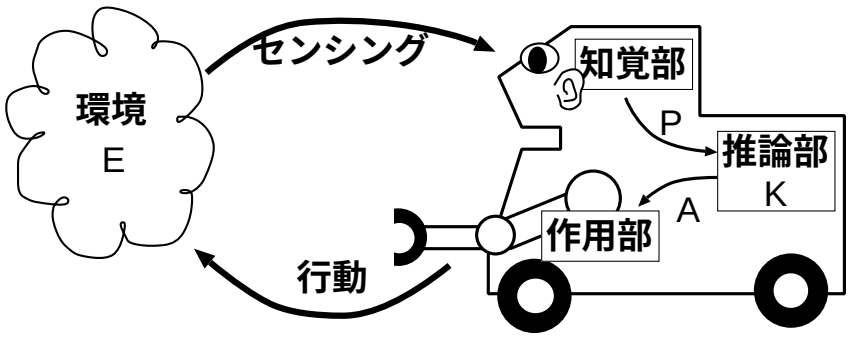


図 9.1: 知的エージェントのモデル図

知覚部： センサを通して外部環境を知覚する部分である．その入力は**外部環境の状態集合** (ここでは簡単のため E と同一視しておく) で，出力は環境のエージェント内での記述である**知覚記述** (percept) の集合 P である．このとき知覚部は，

$$perceive : E \to P$$

なる写像 *perceive* を実現するモジュールである．外部環境の状態はエージェント側から見ると，センサで計測されたセンサ信号で表現されるパターン情報である．逆に，知覚記述は，エージェントのもつ，環境のモデルや環境に存在するオブジェクトのモデルとマッチングすることにより，解釈されたもので，通常はシンボルで表された情報である．この意味から知覚部は，いわゆるパターン認識 (pattern recognition) を行う部分である．

推論部： 知覚部で得られた知覚記述 P を基に，行動の内部記述 (単に**行動記述**と呼ぶ) の集合 A を与えるモジュールである．また，ある意味においては，行動のプラニングを実行する部分と考えても差し支えない．推論部には，何らかの形式的な記述で与えられる**知識ベース** K の存在を仮定し，そこにはドメインやタスクに関する知識が含まれているものとする．
ここでは，エージェントが実行可能な推論のタイプとして

- **熟考型推論** (deliberative reasoning)
- **即応型推論** (reactive reasoning)

を考える．まず，熟考型推論は，知識ベースの関与する推論で，以下の 2 つの写像 *knowledge, deliberate* で特徴付けられる．

$$knowledge : P \times K \to K$$

$$deliberate : P \times K \to A$$

次に，即応型推論は，次の写像 *reactive* で特徴付けられる．

$$reactive : P \to A$$

作用部： 推論部から導き出された行動記述 A と外部環境 E に対して，知的エージェントが作用して外部環境の状態変化を促すモジュールである．作用部の出力は，外部環境に対するエージェントの行動と見なせる．
作用部の機能を形式的に書くと，以下の写像 *effect* となる．

$$effect : E \times A \to E$$

知的エージェントの具体例を考えよう．

- 人間エージェント (人間)：知覚部のセンサとして，目や耳などの感覚器があり，作用部には，手や足など身体の各部が相当する．
- ロボットエージェント (ロボット)：知覚部のセンサには，カメラやレーザレンジファインダが，作用部には，例えばモータで駆動されるタイヤなどが相当する．このエージェントが対象とするのは物理的な環境である．
- ソフトウェアエージェント (Softbot)・知識エージェント (Knowbot)：Softbot とは，自律的なソフトウェアの総称で利用者の代理として動作するものと定義されている．近年，ネットワーク内に存在

するものとして考えられることが通常で，この場合ネットワークエージェントと呼ばれることもある．Knowbot とは，情報フィルタリングをも行うもので，Knowledge Navigator とも呼ばれる [3]．このエージェントは，いわゆるサイバーワールド (cyber world) と呼ばれる電子的な環境を対象とする．このとき，環境における情報表現と知覚・行動記述はいずれも，シンボル情報であるため，知覚部と作用部はあまり意識されないことが多い．

以上は知的エージェントをかなり抽象化したモデルである．これを工学的に実現することが人工知能の究極の目標でもある．ちなみに，このモデルの中で，知覚部がパターン認識 (画像認識，音声認識など)・パターン計測，推論部が人工知能・知識工学，作用部が制御工学・ロボット工学というように，従来は各々の研究分野でほぼ独立に研究が進められてきた．ところが，本当に知的なエージェントを実現するには，各部分を有機的に統合したシステムとして検討考察を深化させねばならないことは言うまでもない．

9.2 エージェントのタイプ分け

本節では，知的エージェントのタイプ分けについて考察する [2,3]．まず，**合理的なエージェント** (rational agent) からスタートする．

合理的なエージェント： 正しい行動をとるエージェント．正しい行動とは，エージェントを最も成功に導く行動である．

合理的なエージェントでは，エージェントのタスク遂行に関する成功を評価する仕組みが必要である．そのために，性能測度 (performance measure) というものが考えられており，それは,「どのように (how)」と「いつ (when)」との両面をもつ．前者は，エージェントがどのように成功するかを決める規範であり，後者は，性能評価のタイミングである．

任意の時点での**合理性**は以下の４つに依存する．

- 性能測度
- 完全な知覚記述系列：エージェントがその時点までに知覚したもの全て
- 環境に関するエージェントの知識
- エージェントが行い得る行動

以上を基に，**理想的かつ合理的なエージェント** (ideal rational agent) の定義を示す．

理想的かつ合理的なエージェント： 完全な知覚記述系列が与えられたとき，性能測度を最大化すると期待される行動を常にとるエージェント．

理想的かつ合理的エージェントは，知覚してきたことに対し，期待される成功が得られるように行動するのである．

また，別の観点から，**自律的エージェント** (autonomous agent) という概念も重要である．

自律的エージェント： 行動選択が，環境に関する組み込まれた知識よりむしろ，自身の経験に依存する割合の高いエージェント．

これは近年，知能ロボットの分野で盛んに研究をされているものである．未知なる環境に関する記述は，完全には得られないので (情報の不完全性)，経験に従いその種の知識を学習して行こうとするエージェントである．いわば，完全な**組み込み** (built-in) 知識をエージェントに事前に与えるのは困難であり，また組み込み知識のみで動作するエージェントは環境の変動に柔軟ではないという立場である．

このエージェントの考え方の背景には，人間の成長の過程や生物の進化の仮定に関する洞察がある．すなわち，最初は組み込まれた反射的な反応しか行えない生物が学習能力により生存環境に適応していくという知見である．同様に，経験の少ないエージェントは初期のわずかばかりの知識に基づき，当初のランダムな試行から始めて，次第に意味のある行動パターンを獲得して行くのである．このように，経験に従い，当初予想もしない挙動を発現させるシステムを**創発システム** (emergent system) といい，最近，研究が活発化している．

最後に，分散協調システムとして着目されている**マルチエージェント** (multi agent) の定義を示す．

マルチエージェント： 複数のエージェントが協調や競合などの相互作用を伴いながら，総体として機能を実現するもの

エージェントの分散方法には，空間分散，機能分散などがあるが，エージェントの個々の機能は比較的単純なものを仮定する場合が多い．

また，マルチエージェントは**均質性**でも分類できる．

均質マルチエージェント： 各エージェントが同一の機能，能力をもつとき，均質 (homogeneous) マルチエージェントと言う．逆に各エージェントが異なる機能，能力をもつとき，非均質 (heterogeneous) マルチエージェントと言う．

9.3 環境

エージェントがインタラクションをとる環境を，Russel の特徴付け [2] に従って分類する．

到達可能－到達不可能： エージェントのセンサが環境の状態を完全に把握できるとき，言い換えると行動を選択するのに適するすべてのアスペクトをセンサが検出可能なとき，環境は到達可能 (accessible) である．そうでないとき到達不可能 (inaccessible) である．

決定的－非決定的： 環境の次の状態が，現在の状態とエージェントによって選択された行動により完全に決定できるとき，環境は決定的 (deterministic) である．そうでないとき非決定的 (nondeterministic) である．到達可能で決定的な環境では，不確実性を考慮する必要はない．

エピソード的－非エピソード的： エージェントの経験が複数のエピソード（エージェントの知覚記述とそれに対応する行動の組）に分割されるとき，環境はエピソード的 (episodic) である．そうでないとき非エピソード的 (nonepisodic) である．エピソード的環境では，行動の質は現在のエピソードに依存し，未来のエピソードは過去のエピソードに依存しない．

静的－動的： エージェントが考えている間に環境が変化し得るとき，環境は動的 (dynamic) である．そうでないとき静的 (static) である．静的環境では，時間の経過への注意，及び行動を決定する間の環境の観測が不要である．環境が時間の経過には変化しないものの，エージェントの行動により環境が変わり得るとき，環境は半動的 (semidynamic) である．

離散的－連続的： 知覚・行動記述の情報表現が有限個に区別できるとき，環境は離散的 (discrete) である．そうでないとき連続的 (continuous) である．

対象とする環境に応じて，エージェントの能力 (具体的にはプログラム) を変更する必要がある．ここで注意すべき点は，エージェントの視点・立場に従って，環境の見かけ上の特徴が変化し得ることである．特に，決定的と非決定的に関してそのことが言える．また，上記の特徴はすべてが独立な軸ではなく相関がある．例えば，ある環境が到達不可能ならば，それは非決定的である．いうまでもなく，もっとも困難な環境は，到達不可能・非エピソード的・動的・連続的である．また現実世界はその複雑性から決定的であるか否かは議論のあるところで，一般には非決定的として扱うことが多い [2].

9.4 エージェントの推論

推論とはエージェント内部での記述の変換とみなすことが可能である．先に掲げた２つのタイプの推論は，いずれも知覚記述を出発点にして，行動記述を得るまでの記述の変化プロセスと位置付けられる．

9.4.1 熟考型推論

熟考型推論は，何段階かの中間的記述を経て，行動記述に到達する推論と要約できる．このとき，従来の人工知能の分野で盛んに検討されてきたシンボルによる推論が主たる役割を果たす．そのためには，何らかの形式的な記述の集合である**知識ベース**が不可欠となる．形式的な記述は，如何なるものでも良く，一般に利用されるものには，一階述語論理式，ルール，フレームなどがある (教科書第４章参照)．

この知識ベースを基に，人間の基本推論パターンである演繹 (deduction)・帰納 (induction)・アブダクション (abduction)，あるいは AI で定式化が図られている類推 (analogical reasoning)，仮説推論 (hypothetical reasoning) などが実行される訳である (教科書第６章参照)．

さて，前節で定義した写像 *knowledge* は，知識ベースを順次更新する写像と捉えることができるが，写像の与え方によって推論のタイプが規定できる．例えば，

X： エージェントは知能を持っている．

Y： 007 はエージェントである．

Z： 007 は知能を持っている．

という知識ベースがあるとすると，X と Y から Z への写像が演繹，Y と Z から X への写像が帰納，X と Z から Y への写像がアブダクションとして特徴付けられる．

ここで，車を運転するエージェントを例に取って熟考型推論を考えてみよう．次のようなルール ($\Longrightarrow$ は “ならば” を表す) からなる知識ベースを仮定する．なお，マッチするルールを前向きに順次適用していくことが演繹推論であることに注意されたい．

- 「工事中」$\Longrightarrow$「車線が減少する」
- 「車線減少」$\Longrightarrow$「渋滞する」
- 「渋滞」$\Longrightarrow$「追突が多い」
- 「追突」$\Longrightarrow$「危険である」
- 「危険」$\Longrightarrow$「アクセルを緩める」

いま，エージェントが観測しているシーン中に案内表示版があるとしよう．このとき，エージェントが表示版の中から「前方工事中」という文字列を認識するならば，知覚記述として「前方工事中」という意味のあるシンボル列を獲得したことになる．このプロセスが知覚部で行われる写像 $perceive : E \to P$ であり,「前方工事中」は P の要素である．

続いて，得られた知覚記述が推論部に引き渡される．推論部は，知識ベースとマッチングを取ることによって，推論を進め，内部の知識記述の変更を繰り返し，最終的に行動記述「アクセルを緩める」($\in A$) を導く．この行動は作用部を通して外部環境 E に影響を及ぼし，“車のスピードが落ちた” 環境へと移行する．

なお，ここで注意すべき点は，エージェントが熟考型推論を施す際に，サブゴールを設定して，知識ベースの組織を変更していることである．例えば，前述の運転エージェントならば,「安全」「高速」「燃費安く」「快適」「スリリング」など，車の運転に対するさまざまなサブゴールを設定して，知識ベースを適応的にスイッチしていると思われる [2]．このことは，写像 *knowledge* がサブゴールに連動して変化することに対応する．

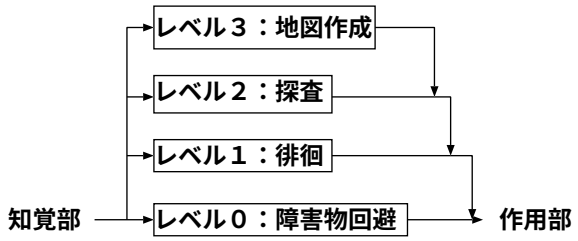


図 9.2: 服属アーキテクチュア

9.4.2 即応型推論

即応型推論は，パターン認識した結果である知覚記述と行動記述との直接的な対応を与えるもの[1] と考えることができる．これを実現する簡便な方法として**即応型ルール** (reactive rule) がある．即応型ルールとは，知覚部の出力である知覚記述をルールの条件部 LHS(Left Hand Side) に，作用部の入力である行動記述をルールの結論部 RHS(Right Hand Side) にしたものである．すなわち，

- 知覚記述 ($\in P$) $\Longrightarrow$ 行動記述 ($\in A$)

が一般的形式であり，例えば運転エージェントでは，

- 「前方に障害物が存在する」$\Longrightarrow$「ハンドルを切る」

などが考えられる．

即応型推論は，環境モデルの完全な記述を必要としないため，センサを持つ移動ロボットにおいて，現在のところ最も頻繁に用いられる枠組みである．Brooks の提唱した**服属アーキテクチュア** (subsumption architecture) [4] はこの範疇に入るものである．従来のアーキテクチュアが，知覚→モデリング→プラニング→実行というように機能的に分割されているのに対し，服属アーキテクチュアは図 9.2 のようにタスクの複雑さに応じて階層的に構成され，上位のレベルが下位のそれを抑制する (服属させる) ように制御する．このアーキテクチュアは迅速性，拡張性，ロバスト性に優れるとされている．しかしながら，以上述べた即応型推論のみで真の意味での知的レベルの高い振舞が実現できるかという点においては，疑問もある．

さて，即応型推論のための方法論として，事例ベース推論，データベース (ルックアップテーブル) の蓄積参照検索などがある．まず，前者と後者の違いは，蓄積していくデータの構造化の度合である．前者がある程度，論理的に構造化して蓄積していくのに対して，後者はパラメータなどをダイレクトに保存蓄積していく．いずれも新しく入ってくる事例に対して，正確に (exact) マッチするものや類似したものを検索して，対応する行動を定める．

9.5 エージェントの学習

9.5.1 知識獲得・知識洗練化

知的エージェントの学習とは，広義には，知識を増大させ，知識・技能を順次習熟させていくプロセスを指すものである．一口で言うならば，経験の積み重ねでタスクの実行能力を向上させるようにすることが学習である．実際には，図 9.1 に示す知覚・知識・行動の内部記述，すなわち，集合 P, K, A の要素を新規に獲得すること，及び，それらの集合間で定義される写像，すなわち *perceive,knowledge,deliberate,reactive*

[1] 即応プラニング (reactive planning) という用語もしばしば使われる．

を新規作成ないしは再定義することに還元される．さて，**機械学習** (machine learning) アルゴリズム (教科書第７章参照) は，**知識獲得** (knowledge acquisition) と**知識洗練化** (knowledge refinement) に大別できるが，上述との対応関係を見ると，前者が各記述の集合の要素を新規獲得することと，写像を新規作成することに相当し，後者が写像の再定義に相当する．

知識獲得の原則は，事例からの**帰納的学習**である．帰納的学習は，教師ありの概念獲得，並びに教師なしの概念クラスタリング・概念形成に分類され，両方について多くのアルゴリズムが提案されてきた．一例を挙げると，即応型ルールをバックプロパゲーション・ニューラルネットで学習する手法が考案されている．これは，P–A 間の写像を教師つき (正解を与えて) で定めようとすることに他ならない．

また，エージェントやロボット向きの手法として，コスト依存型 (cost sensitive) の学習手法がある [5]．これは，センシングのコストなどを考慮に入れて，いかなる規範でセンシングするかについての指針を与えるものである．代表的帰納学習手法の ID3 にセンシングコストを反映させたものが開発されている．

一方，知識洗練化は，**演繹的学習，解析的学習**とも呼ばれ，推論の効率化を目的として，知識ベースへの内省的な扱いに帰着される．シンボルで表現された知識に対する代表的手法には，**説明に基づく学習 EBL**(explanation based learning)，**チャンキング，知識コンパイル**などが挙げられる．例えば 9.4.1 節の運転エージェントでの熟考型推論の例では，知覚記述である「前方工事中」から出発して，５段階の推論チェーンをたどりながら,「アクセルを緩める」という結論に到達する．洗練化は，このチェーンの途中を省略して，“「前方工事中」$\Longrightarrow$「アクセルを緩める」” というルールを作成して，推論のスピードアップを図るものである．

9.5.2 強化学習

近年，エージェント，特に自律ロボットの学習に対する有力手段の一つと目されているものに，**強化学習** (reinforcement learning) がある [6,7]．強化学習は，環境の状態，及び環境から受け取る**報酬** (reward) を入力として，将来受け取る報酬を最大にするような行動を取る戦略を学習するものである．強化学習はいわゆる教師なし学習で，先験的知識を必要としないという好ましい性質を持つ．

以下では，9.1 節の記号を思い出してもらい，強化学習の説明を進める．ここでは，知覚記述集合 P を単に，状態集合と呼び，エージェントは P を識別可能であるとする．ちなみに，P は環境の状態を表すものと考えてよい．強化学習の枠組では，環境は**確率的有限オートマトン**，言い換えるならば，**マルコフ過程**としてモデル化される．すなわち，現在の状態 p と行動 $a(\in A)$ の組から確率的に次の状態 p' へ遷移する．この際の状態遷移確率を $Prob(p,a,p')$ と表し，状態と行動の組に対する報酬を $r(p,a)(>0)$ と記す．報酬はエージェントに強化信号として作用し，エージェントの行動を修正させるもととなる．

いま，状態集合 P から行動集合 A への写像を $Policy$ とするとき，強化学習の問題は，現在から未来に渡る報酬の重み付き総和，すなわち，

$$\sum_{n=0}^{\infty} \gamma^n r_{t+n}$$

を最大にする $Policy$ を発見すること (最適 Policy の発見) と概念的に定義できる．但し，r_t は時刻 t での報酬を表し，γ は割引 (discount) 率と言い，未来の報酬がどの程度，現在の行動に影響を与えるかを制御するパラメータで０と１の間の値を取るのが通常である．

状態遷移確率 $Prob(p,a,p')$，及び報酬 $r(p,a)$ が既知ならば，通常の動的計画法で最適 Policy は求められるが，それらが未知ならば，環境内で試行錯誤しながら，最適な行動を推定する必要がある．このときに用いられる手法の代表的なものに，**Q 学習法** (Q learning) がある．Q 学習法では，状態 p と行動 a の良さを反映する Q 値 $Q(p,a)$ を定義し，試行錯誤によって値を順次更新して行く．更新式は，

$$Q(p,a) \Leftarrow (1-\alpha)Q(p,a) + \alpha(r(p,a) + \gamma \max_{a' \in A} Q(p',a'))$$

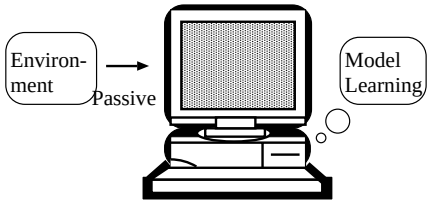


図 9.3: 機械 (コンピュータ) 学習の概念図

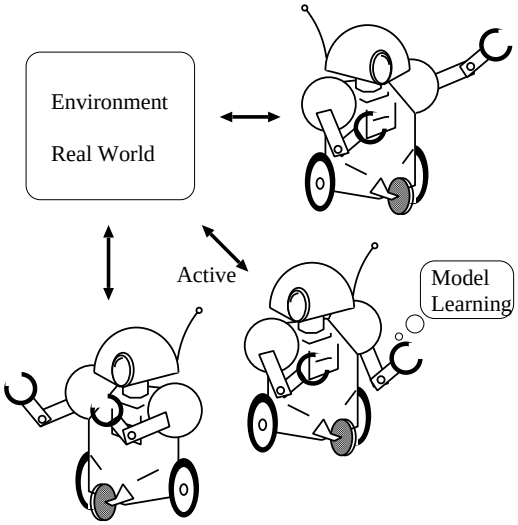


図 9.4: エージェント学習の概念図

で表され，α は学習率で０～１の値を取る．多数の試行により，Q 値が収束した後，各状態において最大の Q 値をもつ行動を選択すれば，最適 Policy となることが証明されている．しかしながら，複雑なタスクでは学習時間が大きくなること，環境 (状態空間) にマルコフ性を要求すること，などの課題も内包している．

Q 学習は最終結果として最大の報酬を得ようと言う最適性を追究する手法である．最適性は，環境を十分に探検 (exploration) することに還元されるため，Q 学習法やその関連手法は，**環境同定型学習** (exploration oriented) と呼ばれることがある．一方，理論的な最適性は望まず，学習途上でも可能な限り継続的に報酬を得ようと言う合理性を追究する手法もある．合理性は，報酬を得た経験 (exploitation) を分析して繰り返すことに還元されるため，**経験強化型学習** (exploitation oriented) と呼ばれることもある [6]．経験強化型学習には，分類子システム (classifier system) などがあるが，詳細は他書に譲る．

9.5.3 エージェント学習のポイント

行動と学習の並行性： これまでの機械学習，特にコンピュータや AI システムに対する記号学習アルゴリズムでは，学習段階と実行 (行動) 段階が明確に区別されていた．要するに，“Execution(Action) after Learning” である (図 9.3 参照).

一方，知的エージェントでは学習済み機械という考え方は通用しない．人間と同等の特徴である，行動と学習の並行性が要請される．すなわち，“Learning during Action” ないしは “Action during

Learning” である (図 9.4 参照)．このための学習アルゴリズムが最低限具備すべき特徴として増進性 (incremental) が挙げられよう．

学習の能動性： 前記の項目とも関連することであるが，従来の機械学習アルゴリズムにおいては，学習者は環境から一方向的に得られる情報を活用する受動的な振舞しか行い得ない．その意味で受動 (passive) 学習である．

エージェント学習においては，先の強化学習の例でも明らかなように，学習者が環境に対して働きかけ，環境から情報を選択的に得るという双方向的な振舞が不可欠となる．このような学習は能動 (active) 学習と呼ばれる．

制限された合理性： エージェントは実環境における行動実体として存在するため，ある種の実時間性は必須条件である．推論やプラニング，そして学習に，どれだけでも時間を掛けても良いとか，いくらでもメモリを利用しても良いとか，といった考え方は不適切である．すなわちエージェントの有する資源 (resource) が有限であることを十分意識して，そういった制約の下で，合理的な振舞を発現させる必要がある[2] ．

ちなみに，これまでの帰納学習理論では，極限同定とか，無数の例題とかいう非現実的な設定が多い．学習においても限られた時間・記憶の中での最良パフォーマンスの追求，すなわち有限の資源における合理性 (bounded rationality) の実現が重要であり，anytime アルゴリズムはその好例である．

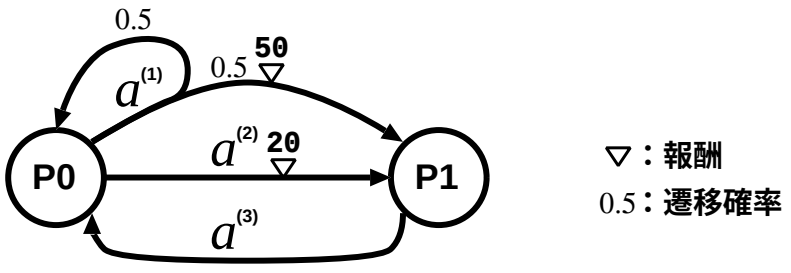
高度な学習環境： 知的エージェント・ロボットに関する既存の学習は，即応型推論をベースに積み上げられているものが大半である．現段階では残念ながら，実時間性への対応から熟考型推論の領域までカバーした学習法はほとんど考察されていない．やはり高度なタスクを遂行するための学習には，熟考型推論，とりわけ非演繹的な推論を視野に入れる必要があろう．

[演習問題]

問 9-1 車を運転するエージェントにおける環境，知覚部，推論部，作用部を具体的に示せ．

問 9-2 次のタスクを行うエージェントが対象とする環境を 9.3 節のタイプに分けよ．
(a) 車の運転　　(b) 車の故障診断　　(c) チェス

問 9-3 下図のような状態遷移図を考える．行動 $a^{(1)}$ によって 0.5 の確率で $P1$ に遷移すると同時に報酬 50 が得られ，0.5 の確率で $P0$ に遷移するが報酬は無い．行動 $a^{(2)}$ によって $P1$ に遷移すると同時に報酬 20 が得られる．行動 $a^{(3)}$ では無条件に $P1$ から $P0$ に遷移する．ここで，報酬の割引率 $\gamma = 1/2$ とするとき，以下の問いに答えよ．



[2] エージェントにおける資源の有限さや物理的実体を考慮して**身体性**という用語が好んで用いられる傾向にある．

(a) n 回の試行錯誤の後の Q 値 Q_n は状態 p_n と行動 a_n に従って次のように定義される．

$$Q_n(p,a) = \begin{cases} (1-\alpha_n)Q_{n-1}(p,a) + \alpha_n\{r_n + \gamma \max_{a'_n}\{Q_{n-1}(p'_n, a'_n)\}\}, & p = p_n, a = a_n \text{のとき} \\ Q_{n-1}(p,a), & \text{それ以外} \end{cases}$$

但し，上式中の α_n は学習率を表し，$0 \leq \alpha_n < 1$ の値をとる．r_n，p'_n，a'_n は各々，状態 p_n で行動 a_n を選択したときの報酬，行動 a_n を選択した後の状態，状態 p'_n での行動を表す．

いま，初期状態 $P0$ から 1 回目の行動 $a^{(1)}$ を実行して $P1$ に遷移し，さらに 2 回目の行動 $a^{(3)}$ を実行して再び $P0$ に戻った場合の $Q_i(P0, a^{(1)})$，$Q_i(P0, a^{(2)})$，$Q_i(P1, a^{(3)})$ $(i = 1, 2)$ の値をそれぞれ求めよ．但し，$Q_0(P0, a^{(1)}) = Q_0(P0, a^{(2)}) = Q_0(P1, a^{(3)}) = 0$，$\alpha_1 = \alpha_2 = 1/2$ とする．

(b) 最適 Policy を実行したときの重み付き総和 $V^*_\infty(p)$ および Q 値の期待値 $Q^*(p,a)$ は以下の式で表される．但し，$r(p,a)$ は状態 p において行動 a を選択したときの報酬の期待値，$Prob(p,a,p')$ は状態 p において行動 a を選択したときに状態 p' になる確率を指す．

$$\begin{aligned} V^*_\infty(p) &= \max_a(r(p,a) + \gamma \sum Prob(p,a,p')V^*_\infty(p')) \\ Q^*(p,a) &= r(p,a) + \gamma \sum Prob(p,a,p')V^*_\infty(p') \end{aligned}$$

このとき問題 (a) の学習率 α_n が条件

$$\sum_n^\infty \alpha_{n(p,a)} = \infty, \text{ かつ} \sum_n^\infty (\alpha_{n(p,a)})^2 < \infty, \quad \text{但し, } n(p,a) \text{ は状態 } p \text{ で行動 } a \text{ が行われた回数}$$

を満たすならば, $n \longrightarrow \infty$ のとき

$$Q_n(p,a) \longrightarrow Q^*(p,a)$$

であることが証明されている．

α_n が上の条件を満たすとき，Q 値の収束値 $Q_\infty(P0, a^{(1)})$，$Q_\infty(P0, a^{(2)})$，$Q_\infty(P1, a^{(3)})$ を求め，最適 Policy を調べよ．

参考文献

[1] M.R.Genesereth and N.J.Nilsson,“Logical Foundations of Artificial Intelligence”, Morgan Kaufmann(1987).

[2] S.Russell and P.Norvig, “Artificial Intelligence, A Modern Approach,” Prentice-Hall(1995).

[3] 石田 亨, “エージェントを考える, ” 人工知能学会誌, Vol.10, No.5, pp.663-667(1995).

[4] R.A.Brooks, “A robust layered control system for a mobile robot,” IEEE Trans Robotics & Automation, Vol.2, No.1, pp.14-23(1986).

[5] 開 一夫, 松原 仁, “機械学習から見たロボット学習,” 日本ロボット学会誌, Vol.13, No.1, pp.5-10(1995).

[6] 山村 雅幸, 宮崎 和光, 小林 重信, “エージェントの学習,” 人工知能学会誌, Vol.10, No.5, pp.683-689(1995).

[7] 浅田 稔, “ロボットの行動獲得のための能動学習, ” 情報処理, Vol.38, No.7, pp.583-588(1997).