目次

第 1 章 はじめに

1.1 ロボットの進化

現在多くの分野でロボットが活躍しており，より身近なものとなってきた[1]．身の回りには工業用ロボットから家庭用ロボットまで様々な場面で様々な機械が見られるようになった．ロボットが現れた初期のころは多くの場合，ロボットはある入力に対して決められた出力が用意されているような単純なものであった．そのため，工場などで利用されるようなアーム等，特定の用途でのみ用いられることが多かった．

近年ではロボットに用いるハードウェアの進化に伴ってより高度な作業が可能になってきた．通信回線を利用した遠隔操作や，人間には作業のできないような環境下での作業，より精密な部品の生産などハードウェアの進化によって出来るようになったことは多くある．

また，ソフトウェア側もより多様な環境に対応できるものへと進化してきている．複雑な環境に対応するための学習機能など様々な研究が現在も活発に行われている．このようなハードウェア・ソフトウェアの進化によってロボットの需要は様々な場面に広がりを見せている（Fig. 1.1）．

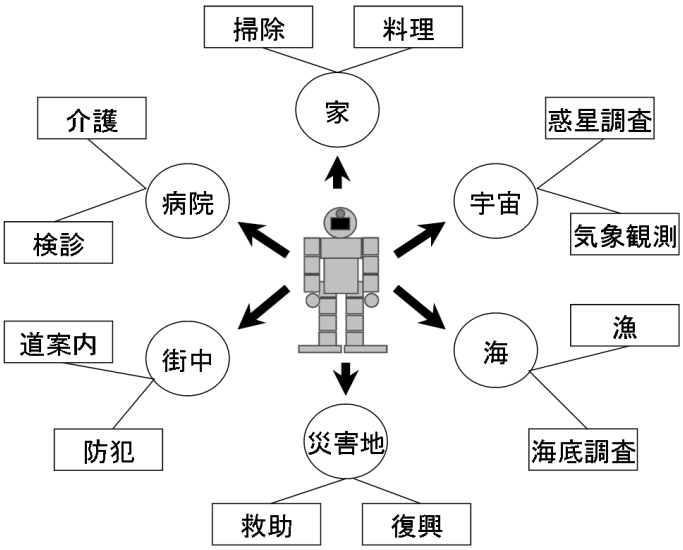


Fig. 1.1：ロボットの需要

1.2 学習による環境への適応

近年では特に学習機能を搭載したロボットが多くみられるようになった．これは実際の環境は非常に複雑であるため，入力から単純に出力を決めるようなロボットでは対応できないためである．また，ロボットを使用する人が望むようなリアクションを学習していく

ロボット等，使用者がより便利だと感じるように学習機能が搭載されることもある．このようにロボットの進化によって様々な場面で活躍する機会が増えたために，ロボットはより多種多様な環境へと適応する必要が出てきた．学習は複雑な環境・多様な要望等に適応するための手段として研究が行われている．人間のようにとまではいかないが，ロボットは学習機能を持つことでより環境へと適応し易くなり，より便利な社会になるために貢献している．こうした学習に関する研究は多く行われており[2]-[5]，様々な分野で活躍している．

身近な学習ロボットの例として，掃除ロボットがある(Fig. 1.2)．掃除ロボットは人間の操作を必要とせず家の中の掃除を行い，掃除が終わったりバッテリーが少なくなったりすると自動で充電スポットまで戻る．このような掃除ロボットにもより効率的になるように学習機能が搭載されている．家のマップを記憶したり，家のどの個所がよく汚れるのかを推測したり，家内をどういった経路で掃除すれば効率良いかを考えたりなど単純に動いているわけではない．また，携帯端末なども使用者の癖を学習する機能が搭載されていることが多い．

Fig. 1.2：掃除ロボット

1.3　学習と認識の関係

こうした学習は通常人間と同様に感覚器官（センサ）を通じて行われる（Fig. 1.3）．センサからの入力によってロボットは自身の置かれている状況や自身の状態を把握し，認識した状況に適した出力を選択する．そしてこの出力に対するフィードバックによって学習を進める．そのため学習においてセンサからの入力は非常に重要になってくる．

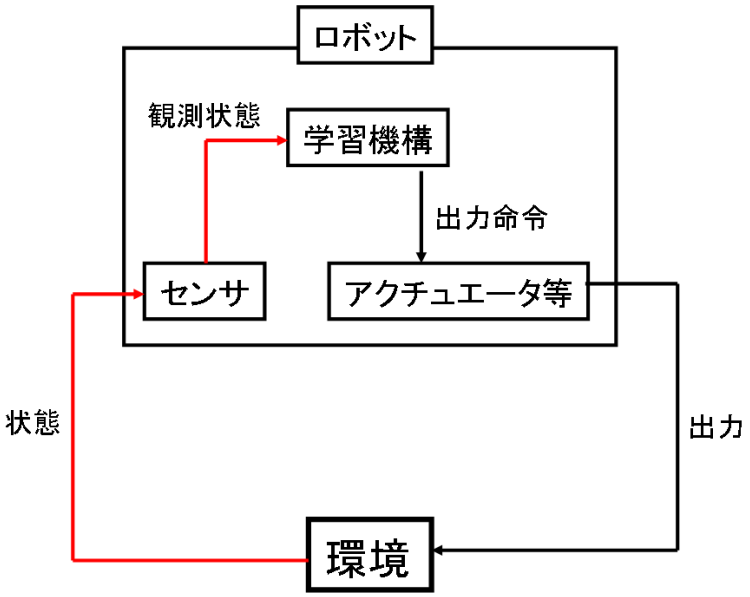


Fig. 1.3：学習と認識の関係

様々な学習の中でも，ロボットの行動学習においてはセンサによる状態認識が特に重要になる．行動学習では自身を含むロボットの周囲の状態に対して適切な行動を見つけていく．そのため細かな状態を認識できた方がより望ましい行動を獲得できる可能性が高くなる．つまり，センサの能力が高ければより適切な学習が可能であるということである．対して，センサの能力が低かった場合には，ロボットの周囲の状態を大まかにしか捉える事が出来ないため，学習が適切に進まない可能性がある[6].

1.4　環境と認識の関係

現在センサの進歩などによってセンサの能力不足が生じるということは少なくなったように思える．しかし実際は，我々の生活環境でロボットの行動学習を行う場合には，高性能のセンサを用いたとしても実環境を正確にトレースすることは難しい．多種多量のセンサを搭載すれば正確に認識することは可能なのかもしれないが，それは現実的ではない．そのため実環境でロボットを扱う場合は学習を行うための状態認識が十分に行われない可能性が常に存在する（Fig. 1.4）．つまり実際の環境とセンサを通したロボットの認識する状態は完全に一致することはなく，この不完全性が問題となる場合がある．この不完全性または環境認識と学習に関しては様々な研究が行われている[7]-[17].

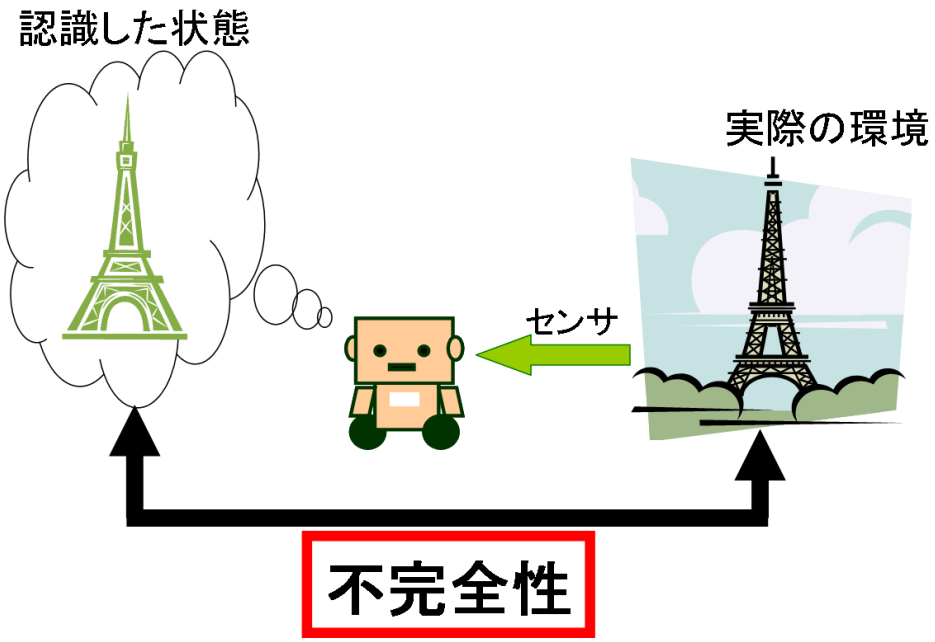


Fig. 1.4：実際の環境とセンサを通したロボットの認識状態

1.5　認識の不完全性がもたらすロボットの行動への影響と問題

1.3 節で述べたように，学習はセンサを通して認識した状態を用いる．そのため認識そのものに不完全性が存在する場合，学習そのものも不完全になってしまう．環境の認識が不完全であると，必要な情報が欠落していたり，本来は全く異なる状況を同じ状態として認識してしまうことがあり得る．本来異なる状況を同じ状態として認識してしまうと，ある行動の結果が毎回異なってしまい学習が進まないことが考えられる．

例えば掃除ロボットの場合，本来は壁にぶつかったことを接触センサ等で認識して，方向を変える動作を取る．しかし，こういったセンサを搭載していなかった場合は，壁にぶつかったことを認識できず，方向を変える動作を取らないため壁にぶつかり続ける．こうなると掃除ロボットは本来の役目（掃除や家のマッピング等）を果たすことができない．

このように認識の不完全性は学習する以前に注目すべき問題を持っている．こういったことは人間にもあてはめることが出来る．目を閉じれば大抵のことが上手く出来なくなるし，耳を塞げば楽器の練習が捗らなくなる．こうした認識の問題はロボットにも起こりうる問題である．センサの能力がロボットに与えられたタスクに対して不十分である場合には，本来区別しなくてはならない状態を区別できなくなる（Fig. 1.5）．この問題がロボットの学習に影響を与えていると考えられる．

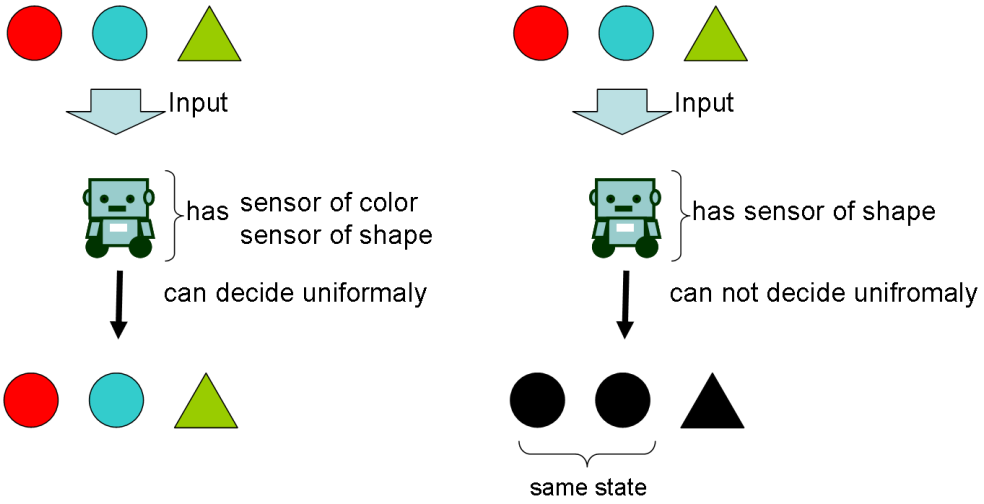


Fig. 1.5：不完全性がもたらす認識の問題

1.6 目的

本論文ではこの不完全性を不完全知覚として学習と不完全知覚の関係に注目する．今回は学習の例として実機に用いられることも多い強化学習を取り上げ，不完全知覚が強化学習にもたらす影響をシミュレーションで見ていく．また，不完全知覚を改善するための手法を提案することを本論文での目的とし，その提案手法の有効性をシミュレーションによって示す．

1.7 アプローチ概要

センサからの入力だけだと不完全知覚となってしまうのであれば，センサからの入力以外の情報を用いて認識できる状態を細分化できれば不完全知覚を改善できると考えられる（Fig. 1.6）．

人間の場合，状態の認識が不完全であっても過去の経験や知識から現在の認識状態をほぼ一意に決定することが出来ます．ただし，これはあくまで過去の経験を基に推測しているにすぎず，実際に正しく環境を認識できているとは限らない．だが，感覚器官からの入力だけではなく自身の経験も用いて認識を行っていることが普通である．

そこで本論文ではロボットの過去の経験情報に注目し，センサの入力のみで現在の状態が一意に特定できない場合は，センサの入力とロボットの過去の経験を用いて認識する手法を考える．ロボットの経験情報を用いることで不完全知覚であっても，より細やかに認識することが出来，認識状態の細分化を行うことが可能になる．

この手法では，ロボットが従来では区別できなかった状態を区別でき，適切に学習を行うことが出来るようになります．

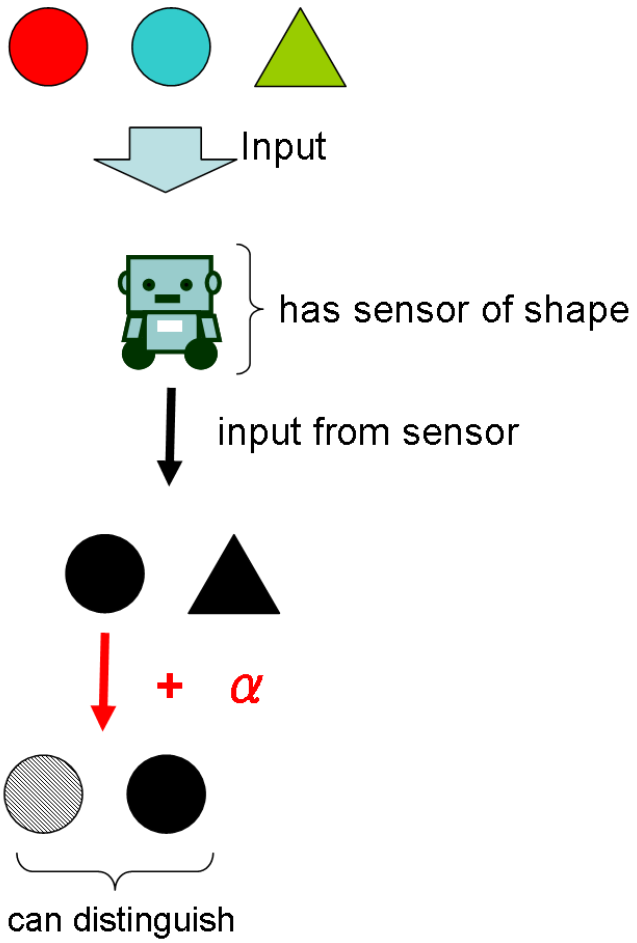


Fig. 1.6 : アプローチ

1.8 論文構成

本論文では２章にて本論文で対象とする不完全知覚について述べ，３章で不完全知覚が原因となる学習時の問題点をシミュレーション実験から考察する．また，４章で不完全知覚を解決するための認識法を提案する．５章で提案手法が不完全知覚に対して有効に働くことをシミュレーションを通して検証する．６章では提案手法における問題点をまとめ，今後の課題として記述する．最後に７章で今後の展開などを述べていく．

第 2 章 不完全知覚

本章ではセンサによる環境の認識について述べる．また，センサによる認識によって引き起こされる不完全知覚の問題について述べる．さらに本論文で対象とする環境のあり方について述べる．

2.1 センサを通した環境認識

本論文ではセンサを用いたロボットを対象としてセンサによる認識と実際の環境，行動学習の関係について注目する．

ロボットが活動を行う環境は複雑であるため，実際にはセンサを通して認識した環境と実環境は一致しない．このような不完全性がありながらも，これまではロボットの活動範囲を予想し，ロボットが与えられたタスクを遂行できるように必要となるセンサを予め搭載することがほとんどである．つまりはある程度限られた環境では不完全性があったとしても大きな問題にはならないと考えられている．

センサを用いて環境を認識する場合はどのような種類のセンサを用いたとしても，ロボットが持つ各種センサの入力の組み合わせでロボットは自身の状態を決定する．例えばロボットが温度センサのみを持っていた場合には，温度そのものが認識した状態となる．また，温度センサと高度センサが搭載されていた場合には温度と高度の組み合わせから現在の状態を決定できる．この時各センサ間には種類や分解能などの違いがあるが，こうした違いはロボットの状態認識において障害にはなることはない．

こうしたセンサを用いた状態認識は，ロボットが認識する状態を観測 $o \in O$（O は観測の集合を表す）とすると式（2.1）で表すことができる．式（2.1）において v_i（i=1,2,…n）はロボットが持つ各センサからの入力を表している．また，n はロボットが持つセンサの数を表している．

$$o = (v_1, v_2, \ldots, v_n) \tag{2.1}$$

センサの数 n が増えたり，センサからの入力 vi がより細かい値で獲得できたりすれば観測 o はより実際の環境に近くなる．しかし，そうすることで観測 o の取りえる数は膨大に増えることになる．観測 o の取りえる数が増えることで様々な状況を認識できるといったメリットがあるが，その分行動学習においては学習にかかる時間も増えるなどのデメリットもあり，必ずしもセンサを増やす・高度なセンサを搭載することが良いことであるとはいえない．

しかし近年ロボットの活動領域が拡大したことに伴って，ロボットを取り巻く環境はよりいっそう複雑化した．複雑になるだけではなくより動的になり従来よりも環境の認識をより正確に行うことが難しくなってきた．また，環境が複雑化・動的になることで人間が

予測しない状況が生まれる可能性も多くなっている.

2.2　センサの能力不足による不完全知覚の発生

ロボットが学習を行う上では環境を完全に認識・把握できることが望ましい．しかし先ほど述べたように実際の環境を完全に認識することは難しい．ロボットが活動する場所をある程度想定出来ていたとしても，予想外の事態が起こりえるため，予めどういったセンサが必要なのかといったことが分からないからである．また，その他の理由として単純にセンサの能力が環境に対して不十分であるために環境を完全に認識できないといったことも考えられる．例えばセンサとして温度計を考えてみると，体温を測るものから気温を図るものまで様々なタイプの温度計が存在する．これらのセンサは全て同じものを計っているが，その精度はそれぞれ異なっている．１℃刻みでしか計れないような場合では 0℃と 0.5℃の違いが認識できない．体温計で言えば 0.5℃の差は結構な差である．こうしたセンサの精度による認識能力の違いはどのセンサにも言うことが出来る．つまりセンサの精度が悪ければ悪いほど観測は実際の環境との差が大きくなることが考えられる.

ここで実際の環境の状態を $s \in S$（S は環境の取りえる状態の集合）とした場合，ある時点 t での実際の環境 s_t とセンサを通した観測 o_t の間には基本的に式(2.2)の関係が成り立つ.

$$O_t \neq S_t \tag{2.2}$$

これは先ほどにも述べたように，実際の環境とセンサを通した状態は一致しないことを示している．ただしこれはセンサの能力不足などにより完全には一致しないということであり，センサは s_t に基づいて o_t を決めるため o_t と s_t の間には何らかの関係が存在する（ただしノイズは考えない）.

こうしたセンサベースの認識においては，ロボットはセンサの能力に合わせて環境の状態をある程度認識できるということになる．そのため実際にはロボットは異なった複数の状態を同じ観測として獲得するということが起こりえる．つまり似たような状態は同じ観測としてとらえてしまう可能性があるということである．一般的にこうした状況は「不完全知覚」と呼ばれる.

具体的に不完全知覚の例を挙げる．明るさを計るセンサの場合を考える．環境の状態として明るさのレベルが１～10 で表され，10 段階で分けられているとする．しかしセンサの能力として２段階でしか認識できない場合（つまり明るいか暗いかの２択の場合），元々10 段階あったものを２段階分まで減らしているため，センサを通した観測には本来の状態を複数含んでいる形になる.ここでセンサは明るさのレベルが 5 以上の時を明るいと観測し，５未満の時を暗いと観測する場合，「明るい」という観測には実際には明るさのレベル 5～10 の 6 つの状態が含まれている．ロボットはこの 6 つの状態を区別することが出来ない．同様に「暗い」という観測には４つの状態が含まれており，区別することが出来ない.

このようにある観測が実際の複数の状態を含んでおり，それらを区別できない状況を不完全知覚と呼び，本研究で注目する問題である．この問題は強化学習[18]において頻繁に取り上げられており[7],[10],[11],[14]-[17]，本論文でも強化学習に基づいて不完全知覚と学習の関係について注目していく．

また,状態集合である S と観測集合 O の間に式(2.3)の関係がある場合には本研究では「完全知覚」と呼ぶ。完全知覚下ではロボットは環境を完全に認識することが出来る．

$$O = S \tag{2.3}$$

2.3 本論文で扱う環境

本論文では強化学習に基づいて不完全知覚の問題を取り上げる．そのため本論文で扱う環境は強化学習で通常用いられる離散化された環境を扱う．実際には環境は連続的であり，どこかで区切れるといったことはない．離散化された環境ではこうした環境をある一定の基準で区切り，状態を定義したものとなっている．強化学習においては基本的な環境のモデルとして MDP[19]がある．また，本研究で扱う環境は強化学習では不完全知覚を考慮したモデルである POMDP[20]に当たる．また，環境そのものは静的な環境を考える．

2.3.1 MDP

強化学習における最も基本的な環境ではマルコフ決定過程（MDP）としてモデル化された環境である．この環境は以下のような特徴を持つ．

- 環境は状態を持ち，ロボットはその状態を完全に認識可能である
- ロボットが行動を行うと環境が確率的に遷移し，環境から報酬を確率的に得られる
- 状態 s'への遷移が，そのときの状態 s と行動 a にのみ依存し，それ以前の状態や行動には関係ない
- どの状態からスタートしたとしても，無限時間経過した後の状態分布確率（どの状態にいるかを表した確率分布）は最初の状態とは無関係である

2.3.1 POMDP

環境の状態を完全には認識できない場合を部分観測マルコフ決定過程（POMDP）というモデルで扱う．マルコフ決定過程の環境では，ロボットによる環境の状態認識は完全であることが仮定されている．しかし，現実ではセンサの能力不足やノイズによって認識した状態に不確実性・不完全性がある場合が多い（Fig. 2.1）．部分観測マルコフ決定過程(POMDP)は，マルコフ決定過程のモデルを拡張し，ロボットの状態認識に不確実性を付加したモデルである．

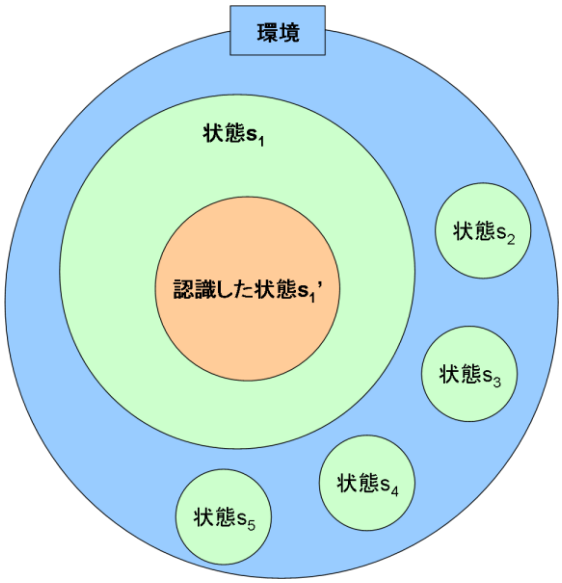


Fig. 2.1：環境の状態と認識した状態の関係

第 3 章 不完全知覚による学習の問題

2 章ではセンサを用いた認識について述べた．さらに不完全知覚といった問題が生じることも述べた．そこで本章では不完全知覚と学習の関係について述べる．様々な学習の中でも特に強化学習について注目し，不完全知覚下での学習への影響をシミュレーションを通して見ていく．

3.1 認識と学習の関係

一般的に学習機構を持つロボット等の場合，センサからの入力に対して適切な出力を学習する．このことはあらゆる学習手法において言えることであるが，本論文では特に強化学習について注目する．まずは強化学習と認識の関係について述べる．また強化学習については付録に簡単にまとめてある．

強化学習では Fig. 3.1 にあるようにロボットはセンサを通じて環境の状態を認識し，それを観測とする．そしてロボットは観測に基づいて適切な行動を学習していく．ＭＤＰ環境下では状態と観測は等しいため，実際の状態に対して学習を行うことが可能である．それに対して不完全知覚を含む POMDP では状態と観測は完全には一致しない．また，ロボットは観測に基づいて学習を進めていく．そのため，本来の状態に対して学習をするわけではないので必ずしも正しい学習結果が得られるとは限らない．

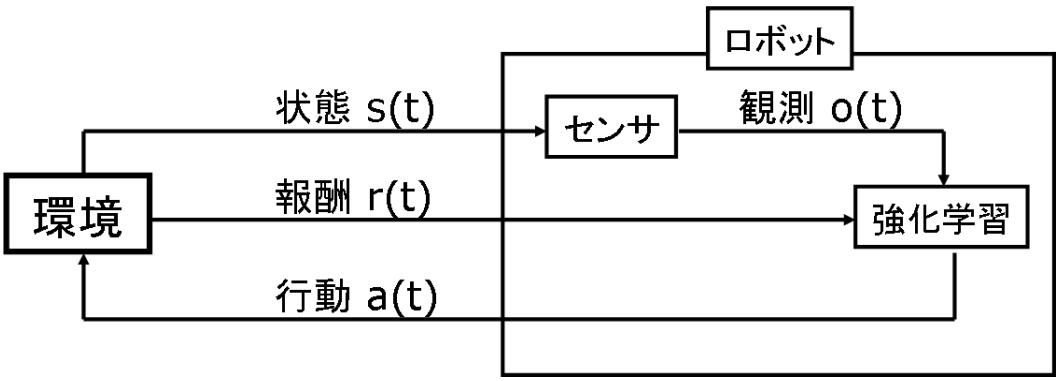


Fig. 3.1：強化学習の流れ

3.2 不完全知覚による学習への影響

強化学習の場合，不完全知覚が学習に必ず悪影響を与えるとは限らない．強化学習ではエージェントがセンサを通して認識した観測を基に学習を行う．そのため各観測ごとに最適な行動を捜すことになる．この時，不完全認識が起こっていると本来異なる複数の状態

に対して一つの最適な行動を探そうとする（Fig. 3.2）．しかし，本来異なる複数の状態について，各状態で最適な行動が異なった場合には不完全知覚を起こしていると最適な行動を学習することは難しい. 例えば Fig. 3.3 のような環境を持つ迷路問題においてスタートの位置が S6 でゴールの位置が S8 であった場合，S3・S4・S5 において最適な行動は S3・S4 は左へ移動に対して S5 は右へ移動となる．そのため不完全知覚を起こしていると学習の段階で右へ移動・左へ移動の競合が起きてしまう．そのため上手く学習できない結果となる．

しかし，不完全知覚であったとしても，最適な行動が同じであるような場合には，学習に影響を及ぼさないと考えられる．Fig. 3.3 で言えば，スタート位置が S6 でゴール位置が S2 であった場合，S3・S4・S5 において最適な行動が左へ移動であるため，学習に影響を与えない．こうした場合にはロボットは正しく学習できる．

そのため不完全知覚で問題が起きる状況は，不完全認識を起こしている複数の状態があり，これら各状態に対する最適な行動が異なる場合であると考えられる．これは最適な行動が同じであればわざわざ状態を区別する必要がなく，学習に支障を与えないためである．

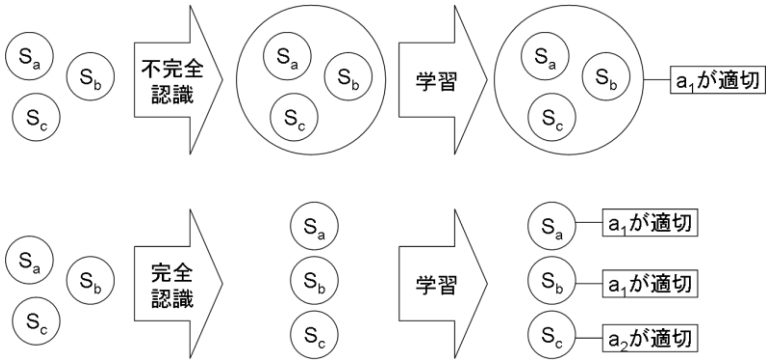


Fig. 3.2：不完全知覚と完全知覚での学習の違い（S は環境の状態，a は行動を表す）

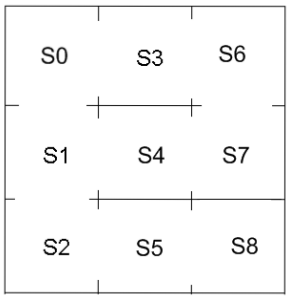


Fig. 3.3：迷路問題における環境（各マスが各状態として割り当てられており，ロボットは各マスを移動することでスタートからゴールまでの道を学習する.）

3.3　不完全知覚エージェントによる学習と学習結果

ここでは不完全知覚が学習に与える実際の影響をシミュレーション実験を通して見ていく. 3.3.1 では実験の目的と概要を述べ, 3.3.2 では実験で用いるエージェントの説明をする. 3.3.3 では実験環境の設定とタスクについて述べ, 3.3.4 ではその他の設定についてまとめている. 3.3.5 で結果を提示し, 3.3.6 で不完全知覚が学習に与える影響について考察する.

3.3.1　実験概要

ここではシミュレーション実験を通して不完全知覚が学習に与える影響を見る. 本実験は迷路問題に対して強化学習を用いて学習を行わせる. ロボットに与えられるタスクはスタートからゴールまでのより身近いルートを見つけ出すことである. この迷路問題に不完全知覚を起こすエージェントと不完全知覚を起こさないエージェントを適用し, 比較を行う. 本実験ではエージェントがゴールするまでを 1 試行とし, エージェントがゴールしたらスタートへと自動的に戻る. 結果は 1 試行ごとの行動数に注目して比較していく. 概要の図を Fig. 3.4 に示す. シミュレーション実験は 2 つ行い, 1 つは不完全知覚が学習に影響を与えないようなタスクの場合, もう 1 つは不完全知覚が学習に影響を与えるようなタスク設定の場合で行う.

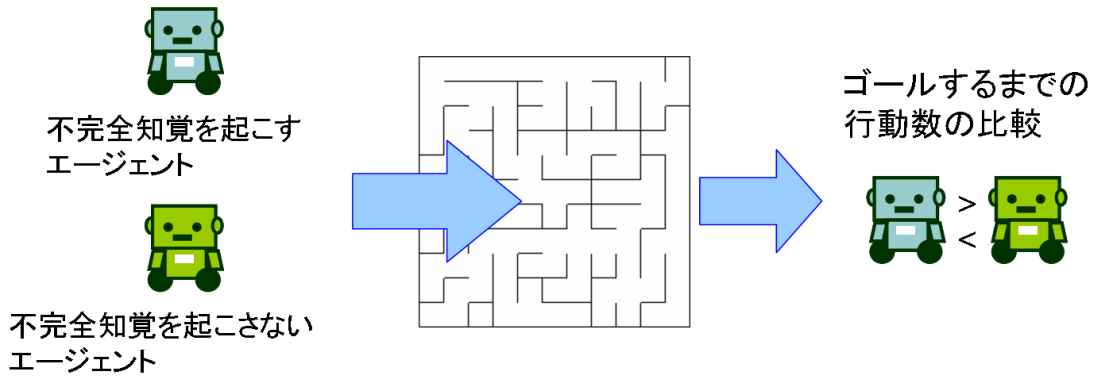


Fig. 3.4: 実験概要

3.3.2　エージェント設定

本実験では 2 体のエージェントを用意する. 1 体は不完全知覚を起こすエージェントであり Agent-A とする. またもう一体は不完全知覚を起こさないエージェントで Agent-B とする.

Agent-A のモデルを Fig. 3.5 に示す. Agent-A は 4 つのセンサを所持している. 各センサは全て同じセンサを用いる. センサは目の前に壁があるかないかの 2 種類の出力を示す. 4 つのセンサを用いることで Agent-A は迷路において自分の周囲の壁の有無を認識できる. そのため観測の種類は Fig. 3.6 に示す 16 種類となっている. これらの観測に対して適切な

行動を学習していく．Agent-A が選択可能な行動は 4 種類｛上移動，右移動，下移動，左移動｝となっており，この 4 つの中から 1 つを選択し実行する．

同様に Agent-B のモデルを Fig. 3.7 に示す．Agent-B は迷路における絶対座標をセンサを通じて得ることが出来る．そのため迷路問題において不完全知覚が起きるようなことは無い．Agent-B の観測の種類は迷路のマスの数（環境の状態の種類数）と同数である．また，認識（センサ）以外の設定は Agent-A と共通である．

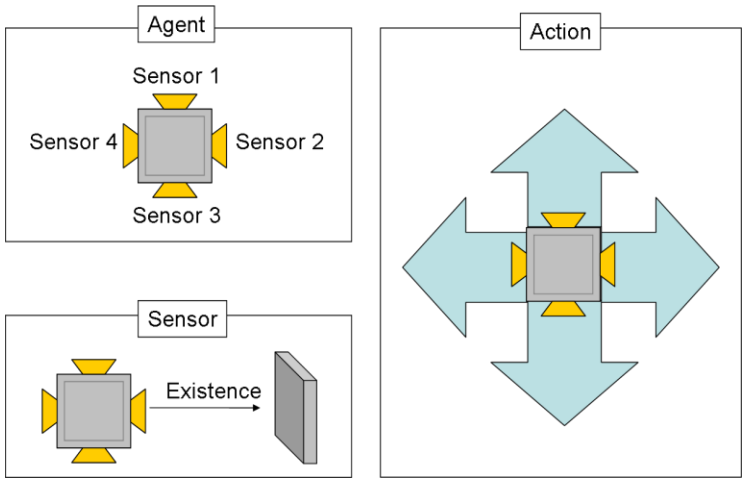


Fig. 3.5：Agent-A のモデル概要

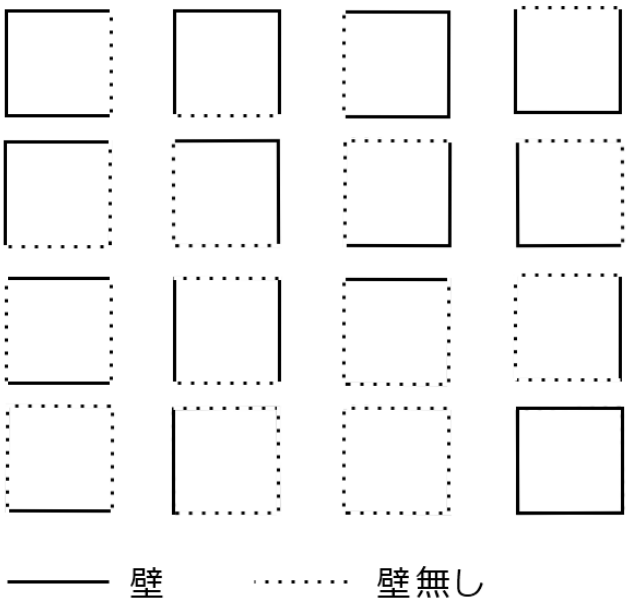


Fig. 3.6：迷路問題における Agent-A による観測の種類

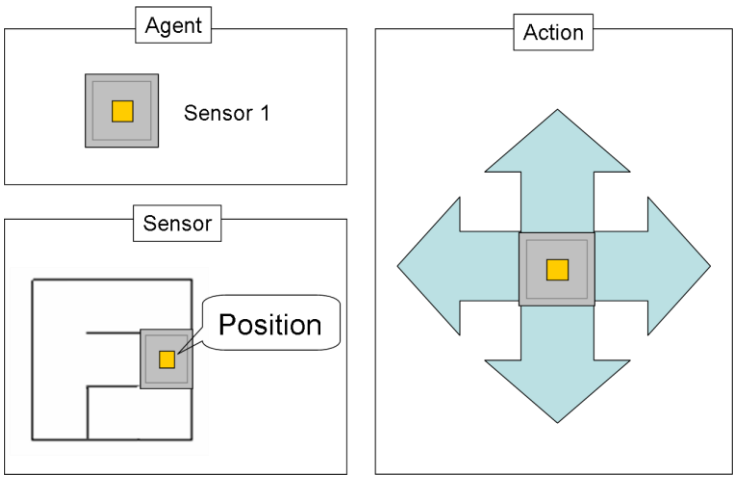


Fig. 3.7：Agent-B のモデル概要

3.3.3　環境とタスク設定

本実験で用いる迷路を Fig. 3.8 に示す．実線が壁である．迷路の大きさは 3×3 マスである．そのためこの迷路の状態の種類は 9 種類ある．この迷路は各マスが左上のマスを原点として右方向に x，下方向に y を座標として持つ．また，この迷路において不完全知覚を引き起こすマスを示した図を Fig. 3.9 に示す．Agent-A において同じ色（橙色）で示されたマスは同じ観測として得られる．この迷路では不完全知覚を起こすマスは座標で言うと（1,0）と（1,1）の 2 か所ある．つまり Agent-A は（1,0）と（1,1）のどちらのマスにいるか区別がつかないということである．

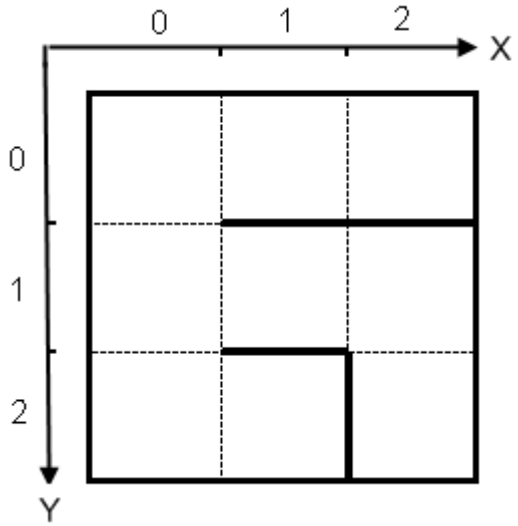


Fig. 3.8：実験で用いた迷路

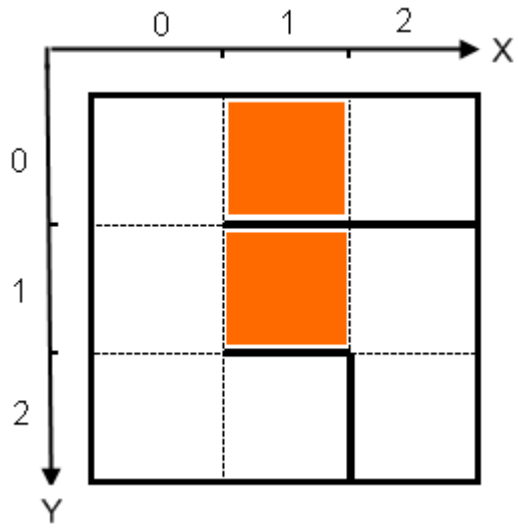


Fig. 3.9：不完全知覚を起こす箇所

この環境において，タスクはスタートからゴールまでのより短いルートを見つけ出すことである．そのためスタート位置とゴール位置を設定することでタスクを設定する．不完全知覚が学習に影響を与えるかどうかはタスクの設定にも左右されるということであったため，今回は２種類のタスクを用意する（２種類のスタートとゴール位置のセットを用いる）．１つ目のタスクはスタートＳを（2,0），ゴールＧを（0,2）とし，タスク１とする．また，２つ目のタスクはスタートＳを（2,0），ゴールＧを（2,2）とし，タスク 2 とする．２つのタスクにおけるスタート位置は共通である．各タスクの迷路図を Fig. 3.10 に示す．

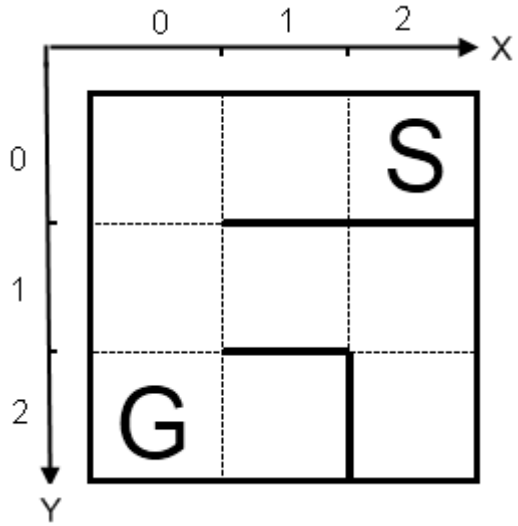


Fig. 3.10 (a) ：タスク１

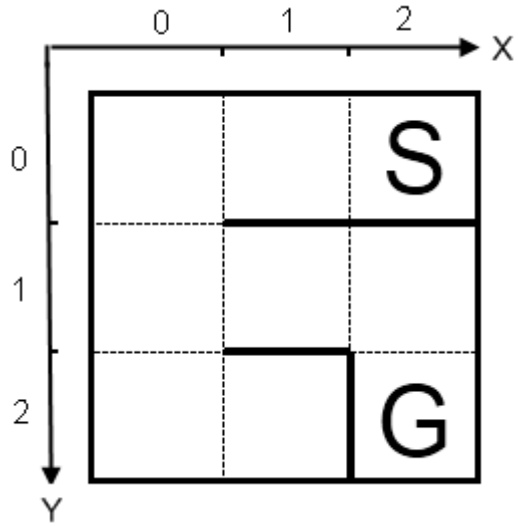


Fig. 3.10 (b) ：タスク２

Fig. 3.10：タスク１とタスク 2 のスタートとゴール位置

3.3.4　その他設定

各エージェントには共通の強化学習手法を適用する．今回用いたのはQ学習と呼ばれる学習手法である．Q学習では式(3.1)によって学習が進められる．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_{t+1}, a_t)] \quad (3.1)$$

また，行動選択手法には ε -greedy を用いる．実験におけるパラメータ，設定のまとめは Table 4.1～4.3 にあるとおりである．

Table 4.1：環境設定

状態数	9
エージェントの行動数	4
スタート位置座標	(2,0)
ゴール位置座標（タスク１）	(0,2)
ゴール位置座標（タスク２）	(2,2)

Table 4.2：学習手法設定

学習手法	Q学習
試行数	30
報酬（ゴール位置のみ）	100
Q値の初期値	0.001
α	0.5
γ	0.7

Table 4.3：行動選択手法設定

行動選択手法	ε -greedy
ε	0.05

3.3.5　実験結果

タスク１の実験結果を Fig. 3.11 に示す．グラフは各試行ごとに見る各エージェントがゴールするまでの行動数である． Agent-A, B はそれぞれ不完全知覚を起こす可能性のあるエージェント，不完全知覚を起こさないエージェントである．X軸に試行数を取ってあり，Trial で表されている．Y軸はゴールするまでにかかった行動数（1行動を１として数える）を表しており，Action で表される．このグラフにおいて下側であればあるほどゴールまで

にかかった行動数が少ないことを表している．つまりグラフが右肩下がりであれば学習がスムーズに行われていると捉えることが出来る．同様にタスク 2 の実験結果を Fig. 3.12 に示す．

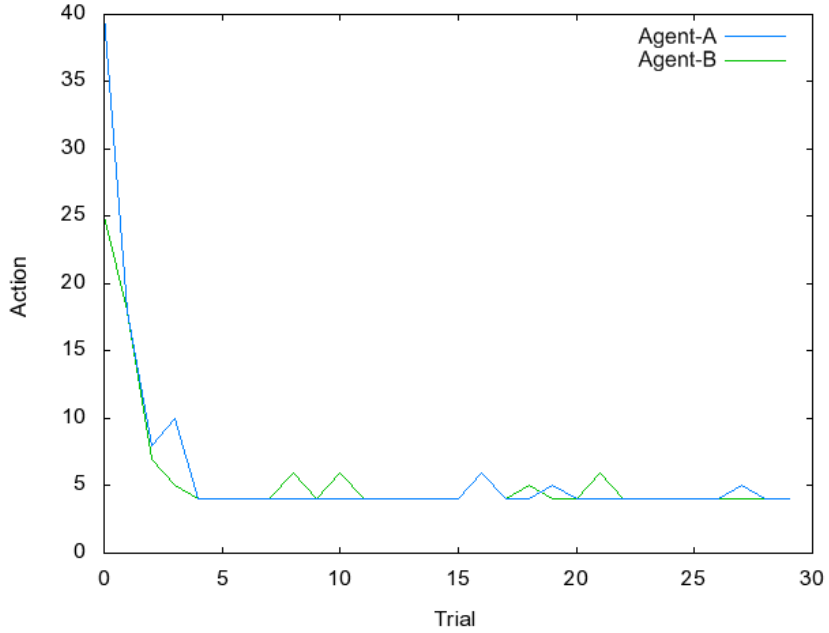


Fig. 3.11：タスク 1 の実験結果（各試行におけるゴールするまでの行動数の比較）

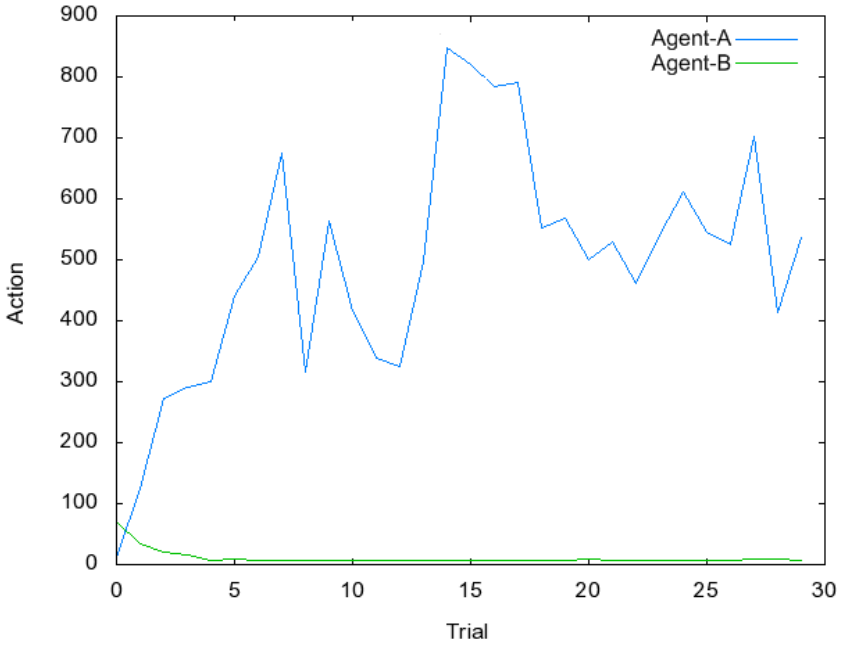


Fig. 3.12：タスク 2 の実験結果（各試行におけるゴールするまでの行動数の比較）

3.3.6 考察

2 つのグラフを見ると，その結果に大きな差が生まれたことが分かる．タスク 1 の場合もタスク 2 の場合も不完全知覚は生じるのに学習が進む場合と進まない場合の 2 種類が存在する．この理由はタスクにあると考えることが出来る．タスク 1 においてゴールの位置を考慮すると不完全知覚を起こすポイント（Fig. 3.9 の橙色のマス）では共通の最適な行動を持つことが分かる．つまり 2 つあるマスのうち，どちらかで最適な行動を学べば両方にその行動をあてはめることが出来るのである．そのため学習上では不完全知覚が問題にならずにいる．

それに対してタスク 2 ではゴールの位置を考慮すると．不完全知覚を起こすポイント（Fig. 3.9 の橙色のマス）では別々の最適な行動（座標(1,0)だと左移動で座標(1,1)だと右移動である）となっているために学習が進まない．どちらかのマスで最適な行動を学習出来た場合には両方のマスでその行動を取ろうとするが，片方はお門違いの方向へ進んでしまう．そのため試行数を重ねてもゴールするまでの行動数は一向に収束せず，学習が進まないのである．全く学習が進まないだけではなく，どんどん悪い方へと傾向が移っていることも見て取れる．これは今回の実験設定に依存する可能性はあるが（遅延報酬型の強化学習等），どちらにせよ今述べたことが原因として学習が進まない傾向は現れる．

このように不完全知覚が必ず学習に影響を与えるとは限らない．しかし，不完全知覚が生じる以上学習に影響が無いとも言い切れない．さらに，不完全知覚が学習に与える影響はかなり大きいことも分かった．そのため学習における状態認識は重要度の高い問題であると考えられる．本論文ではこの不完全知覚そのものに注目し，学習以前の認識の段階での解決を図ることを考える．4 章では不完全知覚に対して有効と考えられる手法を提案し，その具体的な手法について述べていく．

第 4 章　強化学習におけるロボットの経験情報を用いた不完全知覚の改善

3 章では不完全知覚が学習に与える影響をシミュレーションを通して述べた．特に強化学習に注目し，離散化された環境下での認識の問題に注目した．不完全知覚下では状態が完全に観測できないため，学習に悪影響を与えることが分かった．また，不完全知覚でも学習に影響を与えない場合も判明した．そこで 3 章での考察を基に 4 章では離散化された環境下において，不完全知覚を改善するための認識方法を提案する．

4.1　不完全知覚改善のための経験情報

センサを通した状態認識で不完全知覚を引き起こす原因としては以下の事柄が考えられる．

・センサの能力不足（搭載しているセンサの数や識別能力等）

・現在のセンサ値のみの利用

センサの能力不足についてはハードウェアの問題として捉えると，よりよいセンサを搭載したり，センサの数を増やしたりすることで不完全知覚を改善することが出来ると考えられる．しかし，センサの能力を挙げたとしても実際には予測できない状況が生まれることによって不完全知覚が起きる可能性は大いにある．

そこで本論文では「現在のセンサ値のみの利用」に注目する．通常は現在のセンサ値のみから観測を行っている．しかしそれではセンサの能力に依存してしまい，不完全知覚を引き起こす要因になっていると考えられる．

人間の場合もロボットと同様に，目や耳等の感覚器官からの情報を基に現在の状態を認識する．当然，人間の場合も感覚器官からだけの情報では不完全知覚が生じる．例えば目の前に果物があったとする．この果物を見た時人間は「目の前に果物がある」としか認識しない．しかし，この果物が 1 週間も前から見続けてきた場合，「目の前に腐っていそうな果物がある」と認識することが出来る．前者と後者では人間の取る行動に大きく影響を与える．このように人間の場合は自身の経験を基に現在の状態をより細かく認識している．つまり経験情報を用いることで認識している状態を細分化し，センサの能力で認識できる限界以上に認識できる状態を増やしている．

そこで本論文ではロボット自身の経験に注目する．経験情報を用いることで，より細かい状態の認識を目指す．強化学習の場合，過去の経験は観測と行動の 2 つが考えられる．この過去の観測と行動に注目し，不完全知覚の改善を図る．

4.2　経験情報を用いた観測の細分化

センサからの入力だけでは不完全知覚が生じてしまう．そこでロボット自身の経験情報を用いて，観測を細分化することを提案する（Fig. 4.1）．今回は自身の経験情報として直前の観測と行動について取り上げる．また，全ての観測に対して細分化を行うのではなく，不完全知覚が起きていると考えられる観測に関してのみ細分化を行うことを考える．つまりある観測が不完全知覚かどうかをロボット自身が判定し，不完全知覚出会った場合にはその観測に対して直前の観測と行動を用いて細分化を行っていく．このようにすることで無駄なく観測の細分化が行えると考える．

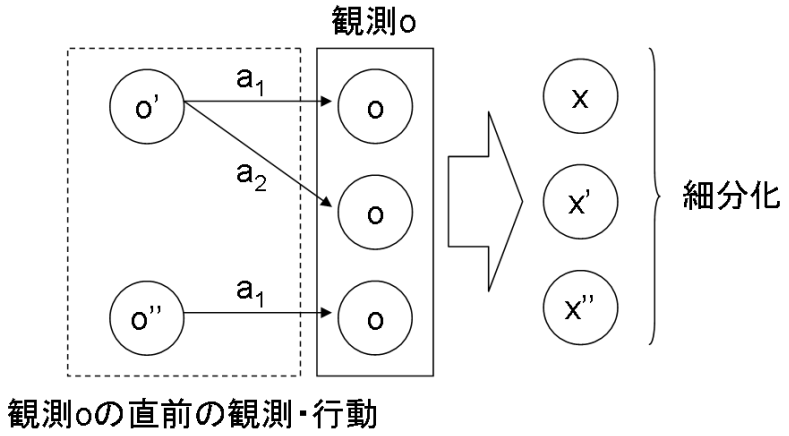


Fig. 4.1：アプローチ

よって注目すべき点は「不完全知覚かどうかの判断」，「不完全知覚である観測の細分化」の 2 点である．3 章で述べたことだが，不完全知覚下で問題が生じる場合は不完全知覚を起こしている複数の状態が異なる最適な行動を持っている場合であった．つまりある観測が最適な行動を複数持っていた場合にその観測は不完全知覚であり，細分化するべき対象として捉える事が出来る．しかし，最適な行動とは学習の結果から導き出せるものだったり，そもそも最適な行動が学習では見つからない可能性もある．そこで最適な行動に注目するのではなく，ある観測が 1 つの行動で複数の異なる結果（異なる観測が得られる）が生まれる場合に注目する（Fig. 4.2）．これは不完全知覚が複数の状態を含んでいることから考えている．複数の状態が含まれているならば，同じ行動の結果が異なる結果を生み出す可能性は高い．この方法で不完全知覚かどうかを判断する場合，ある観測に対して同じ行動を取らなければならない．そのため不完全知覚かどうかを判断する材料としてここでも経験情報を扱う．つまり，ある観測の行動の結果を過去の経験と比較することでその観測が不完全知覚かどうかを判断する．

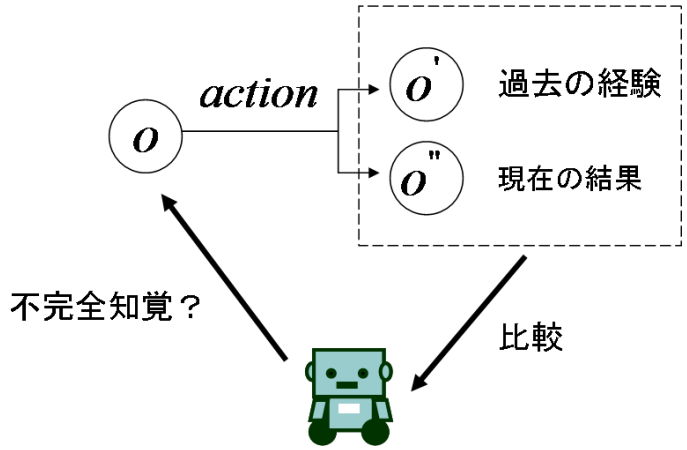


Fig. 4.2：不完全知覚の判定アプローチ

さらに，不完全知覚と判定された観測はその直前の観測と行動を用いて観測を細分化する（Fig. 4.3）．細分化の際，観測そのものと，観測＋直前の観測・行動の 2 つへと細分化する．つまり，何時もの観測と特殊な場合（過去の経験を含めた）の観測に細分化される．そして，観測＋直前の観測・行動をロボットは「状態知識」として記憶することで，その後の認識にこの知識を用いることが可能になる．

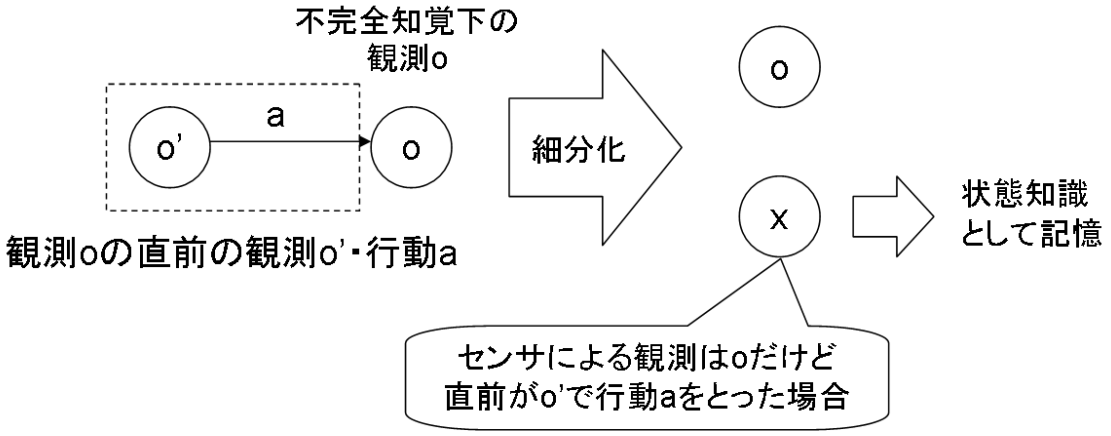


Fig. 4.3：不完全知覚下の観測の細分化アプローチ

また，細分化され新たに増えた分の状態知識はそれ自身さらに細分化されることが可能である．細分化の結果まだ不完全近くであると判断された場合はさらに細分化を繰り返し行うことが出来る．

4.3　観測細分化を利用した状態認識の概要

ここでは，4.2 節で述べたアプローチを基に不完全知覚に対応するための状態認識法について述べる．

4.3.1　概要

提案する状態認識では，不完全知覚であると判定された観測に対して細分化を行い，細分化した時の情報を基に現在の状態が決定される．提案手法では最終的に決定された状態を「認識状態」として$\hat{o}$で表す．つまり，通常はセンサを通して S は O へマッピングされるが，提案手法によってさらに O は$\hat{O}$（認識状態の集合）へとマッピングされる．提案手法は大きく分けて以下の 4 つのモジュールから構成される．

・状態認識部
・経験情報蓄積部
・不完全知覚判定部
・細分化部

さらにロボットは提案手法で用いる以下の 2 つの知識を有する．

・経験知識 E
・状態知識 X

経験知識 E は過去の経験の集合である．また，状態知識 X は細分化の時に作られた状態知識の集合となっている．

「状態認識部」では現在の観測 o（つまり現在のセンサからの入力）とロボットが持つ状態知識 X を用いて現在の認識状態$\hat{o}$を決定する．この状態知識は「観測細分化部」によって作り出される．「細分化部」では「不完全知覚判定部」において不完全知覚と判定された認識状態$\hat{o}$を細分化し，状態知識 x∈X を作成する．これらモジュールと知識の関係図を Fig. 4.4 に示す．また，観測集合 O と認識状態集合と状態知識集合 X の関係を Fig. 4.5 に示す．

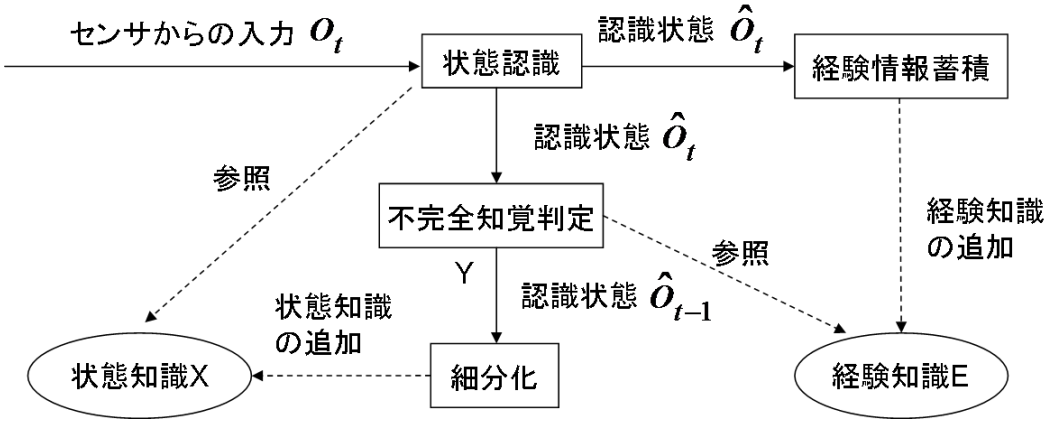


Fig. 4.4：提案手法における各モジュール間の関係

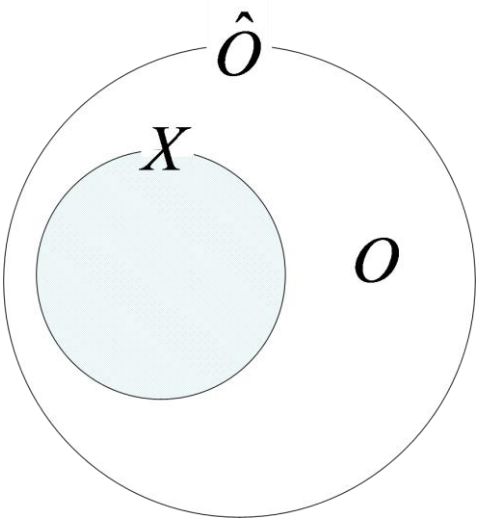


Fig. 4.5：観測と認識状態と状態知識の関係を表したベン図

4.3.2　流れ

これらのモジュールを用いて細分化を考える際，実際に細分化される対象となるものはロボットから見て 1 時刻前の認識状態となる（Fig. 4.5）．その理由は，「不完全知覚判定部」において不完全知覚かどうかを判定する対象が 1 時刻前の認識状態であるからである．つまり，ロボットはある時刻 t において，その時点での 1 時刻前の認識状態に対して不完全知覚かどうか判断し，もしも 1 時刻前の認識状態が不完全知覚ならば 2 時刻前の認識状態と行動を用いて状態知識を作り出し，その後の認識に使えるように保持する．状態認識から細分化までの流れを時刻を含めた形で Fig. 4.6 にまとめる．

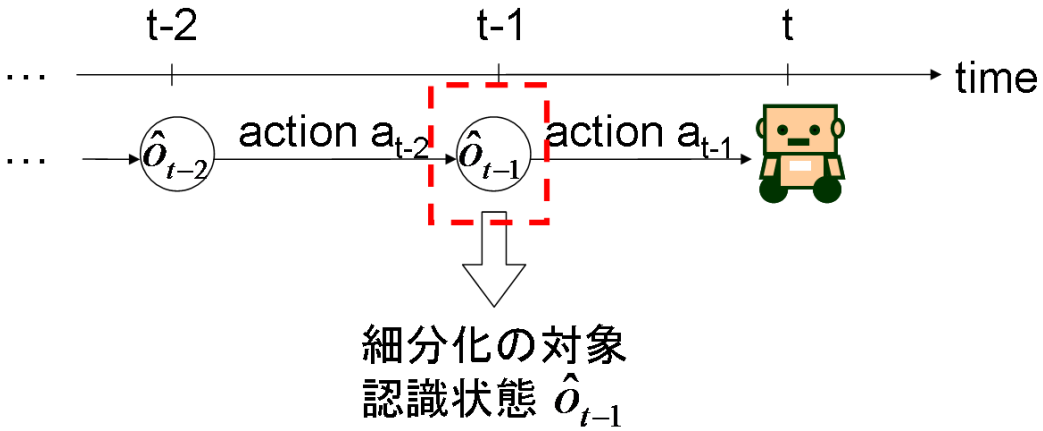


Fig. 4.5：細分化の対象

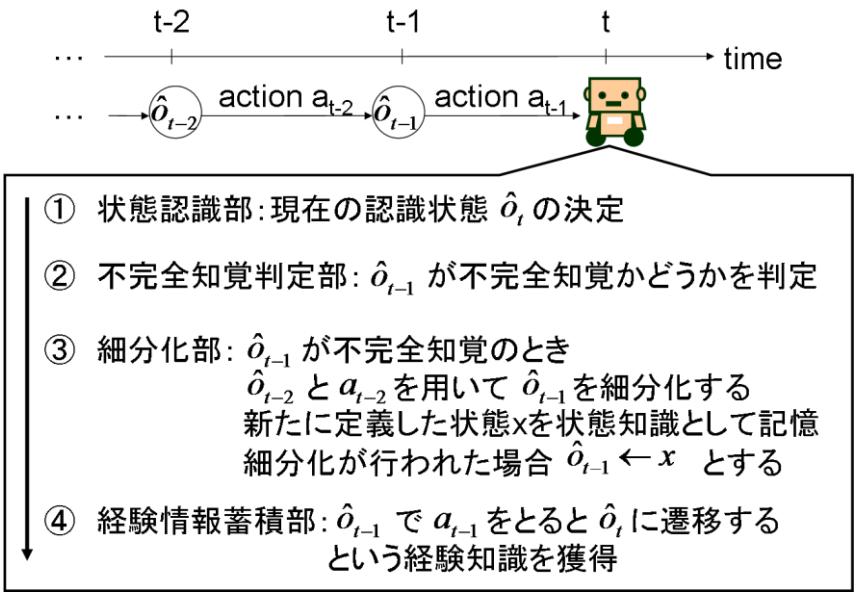


Fig. 4.6：提案手法の流れ

以降の節では各モジュールと 2 つの知識の定義について具体的に述べていく．まず 4.4 節で 2 つの知識の定義について述べる．以降 4.5 節から順に「状態認識部」・「不完全知覚判定部」・「細分化部」・「経験情報蓄積部」について述べていく．また，4.9 節では強化学習に適用した場合のシステムの流れと強化学習の兼ね合いを述べていく．

4.4　提案手法で用いる 2 種類の知識の定義

ここでは提案手法で用いる「経験知識」と「状態知識」の 2 つについて具体的に定義する．まずは経験知識，次に状態知識について定義する．

4.4.1　経験知識の定義

これまでに述べてきたように経験知識とはロボット自身の経験を知識化したものである．本論文では強化学習に注目していることから，ロボットの認識した状態とその時に取った行動に注目して経験を知識化する．そこでロボットがある時点 t における経験知識 $e_t \in E$ を以下の式（4.1）で定義する．$\hat{o}_{t-1}$ は時点 t-1 における認識状態であり，a_{t-1} は時点 t-1 における行動を表している．また $\hat{o}_t$ は時点 t におけるロボットが認識した認識状態を表す．つまりこの知識は，ある時点である行動を取ったらある状態へ移ったことを表した知識となっている．

$$e_t = (\hat{o}_{t-1}, a_{t-1}, \hat{o}_t) \tag{4.1}$$

4.4.2　状態知識の定義

状態知識はロボットが細分化を行った際に特殊な状況として捉えるために作り出す知識である．そのため状態認識部においてこの知識を用いることで，より細かな状態認識が可能となっている．状態知識はある認識状態に対して特定の過去を持つものを知識として扱っている．そのためある状態知識 x∈X は式（4.2）の形で定義する．$\hat{o}$ はある特定の認識状態を表し，$\hat{o}_p$，a_p は $\hat{o}$ の直前の認識状態と行動を表している．状態知識には時間の概念は存在しない．この知識はあくまである認識状態には直前が特定の認識状態と行動であった場合にそれを知識化したものである．

$$x = (\hat{o}, \hat{o}_p, a_p) \tag{4.2}$$

4.5　状態認識部

ここでは状態認識部での具体的なアルゴリズムについて記述する．状態認識部では現在の（時刻 t での）センサからの入力である観測 o_t と状態知識 X を用いて現在の認識状態 $\hat{o}_t$ を決定する（Fig. 4.7）．またこの時 t-1 時刻の認識状態 $\hat{o}_{t-1}$ と行動 a_{t-1} も用いる．具体的にはロボットは現在の観測 o_t に対して直前の認識状態と行動を含む状態知識を探す．この時ロボットが持つ状態知識集合の中に探していたものがあった場合はそれを現在の認識状態として決定する．もし，見つからなかった場合には観測がそのまま認識状態として決定される．具体的には以下の手続きで状態の認識が行われる．

1) 現在の観測を o_t とし，直前の認識状態と行動をそれぞれ $\hat{o}_{t-1}$，a_{t-1} とする

2) 状態知識 X の中に $(o_t, \hat{o}_{t-1}, a_{t-1})$ となる x があるか検索

3) もし $x \in X$ （x が存在する）場合，$o_t \leftarrow x$ として手順 2 へ戻る

4) もし $x \notin X$ （x が存在しない）場合，$\hat{o}_t \leftarrow o_t$ として現在の認識状態を決定

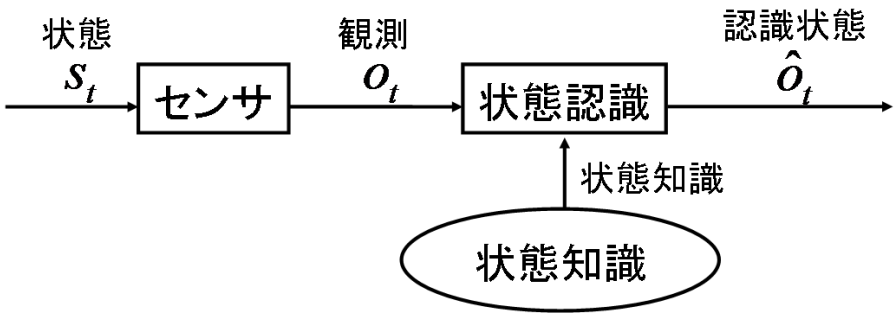


Fig. 4.7：状態認識の概要

4.6　不完全知覚判定部

不完全知覚判定部ではある時点 t においてその直前の認識状態$\hat{o}_{t-1}$を不完全知覚かどうか判断する（Fig. 4.8）. その判断方法は過去の経験と現在の経験を比較することによって判断する．$\hat{o}_{t-1}$において取った行動a_{t-1}を過去にも経験しているかどうかを経験知識を通して探し, その時の結果と現在の結果を比較することで不完全知覚かどうか判定を行う（Fig. 4.9）. つまり過去に同じ状況で同じ行動を取っているのにその時と得られた結果が違うかどうかを見ることで判定する.

この不完全知覚判定部で不完全知覚と決定された場合のみ細分化部で認識状態の細分化が行われる. つまりある時点 t で不完全知覚ではないと判定された場合には時点 t においては細分化部は一切の働きをしない．不完全知覚判定部は以下に示す手順で行われる.

1) 現在の認識状態を$\hat{o}_t$とし，直前の認識状態を$\hat{o}_{t-1}$，直前の行動をa_{t-1}とする

2) 経験知識 E の中から直前の認識状態と行動に関する知識$(\hat{o}_{t-1}, a_{t-1}, *)$を検索

3) 見つかった知識を$(\hat{o}_{t-1}, a_{t-1}, \hat{o}'_t)$とする

4) もし$\hat{o}_t \neq \hat{o}'_t$ならば$\hat{o}_{t-1}$を不完全知覚と判定し，細分化を行う

5) もし$\hat{o}_t = \hat{o}'_t$ならば経験知識 E 内全ての知識を検索し終わるまで手順２へ戻る

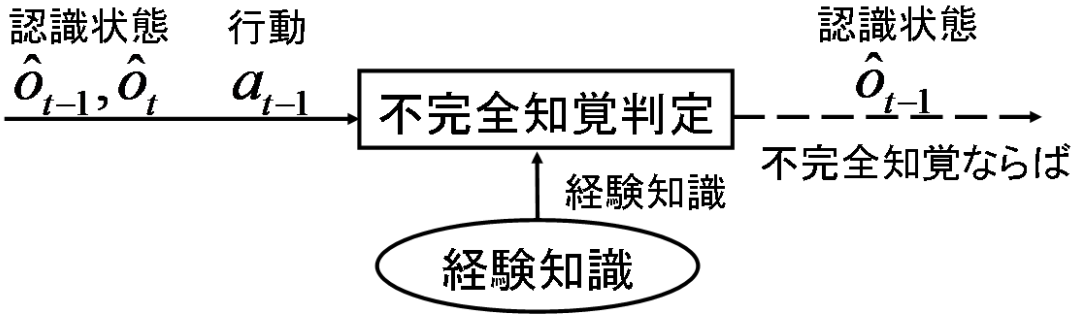


Fig. 4.8：不完全知覚判断部の概要

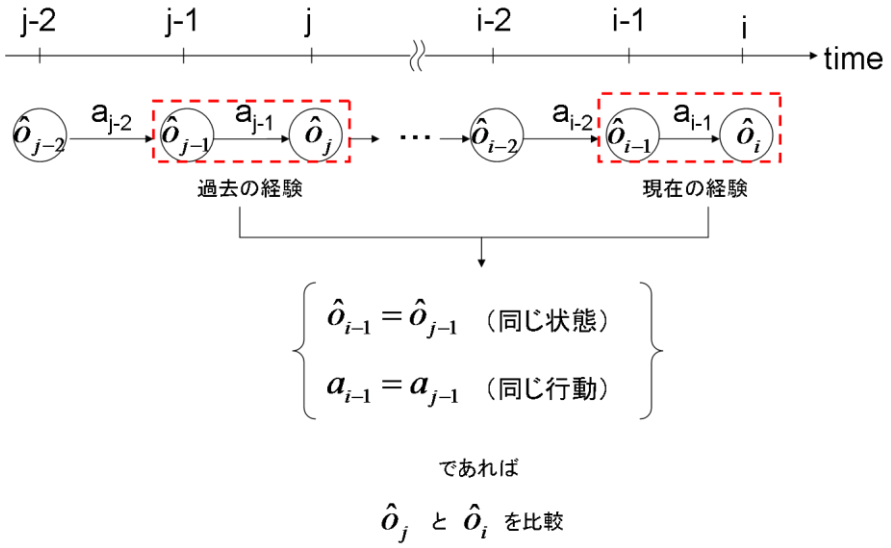


Fig. 4.9：不完全知覚判断の方法

4.7　細分化部

細分化部では不完全知覚判定部において不完全知覚であると判定された認識状態 $\hat{o}_{t-1}$ に対して細分化を行う（Fig. 4.10）．細分化は $\hat{o}_{t-1}$ そのものとそうでない x に分割される．x は状態知識として 4.4.2 項で挙げた式（4.2）の形式で新たに定義される．ロボットはこの新たに定義した状態知識 x を記憶することで, その後の状態認識で使うことが可能になる．具体的な手順は以下のとおりである．

1) 細分化対象を $\hat{o}_{t-1}$ とし，対象の直前の認識状態を $\hat{o}_{t-2}$，直前の行動を a_{t-2} とする
2) $\hat{o}_{t-1}$ を $\hat{o}_{t-1}$ と x に分け，新たな状態知識 x を $x=(\hat{o}_{t-1},\hat{o}_{t-2},a_{t-2})$ として定義
3) ロボットは $x \in X$ とすることで新たに定義した状態知識を保有する
4) $\hat{o}_{t-1} \leftarrow x$ とする

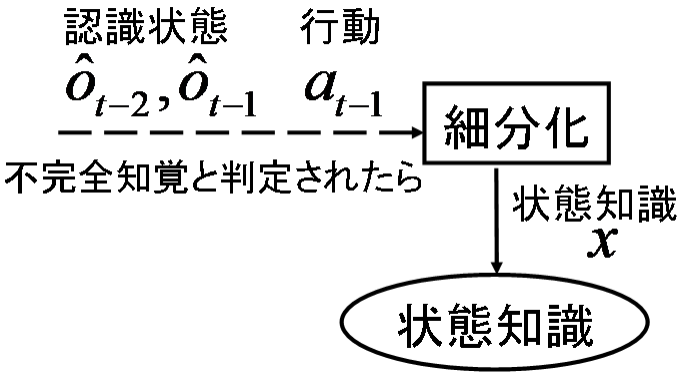


Fig. 4.10：細分化部の概要

4.8 経験情報蓄積部

ここでは経験情報蓄積部について具体的に記述する．経験知識自体は 4.4.1 項で説明したとおり式（4.1）で表される．経験知識はその時の認識状態と直前の認識状態と直前の行動について知識化する（Fig. 4.11）．ある時点 t ではロボットは以下の手順で経験情報を知識として獲得する．この時もし $\hat{o}_{t-1}$ が細分化されていた場合には細分化された方の認識が適用される．

1) 現在の認識状態を $\hat{o}_t$ とし，直前の認識状態を $\hat{o}_{t-1}$，直前の行動を a_{t-1} とする

2) 現在の経験知識を e_t とし，$e_t = (\hat{o}_{t-1}, a_{t-1}, \hat{o}_t)$ として知識化を行う

3) ロボットは $e_t \in E$ として知識を獲得する

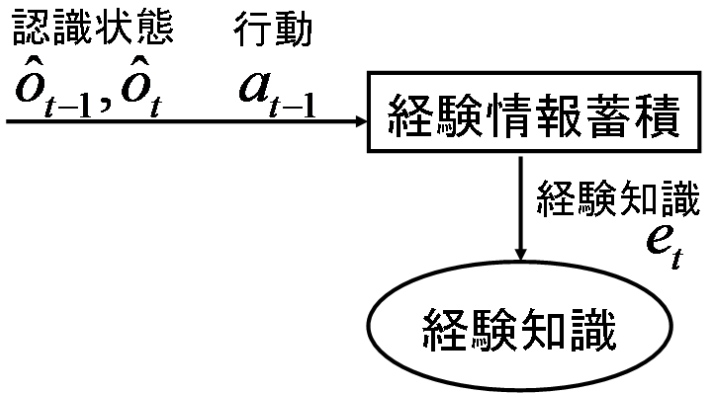


Fig. 4.11：経験情報蓄積部の概要

4.9 提案手法の強化学習への適用

提案手法を強化学習へ適用する場合のシステム概要図は Fig. 4.12 のようになる．最終的には提案手法で決定した認識状態 $\hat{o}_t$ に基づいて学習を行う．提案手法は認識における手法になっているため，基本的に強化学習とは独立に働き，学習手法に何らかの変更を与えることはない．しかし，提案システムでは行動を重ねることで認識できる状態が増えていくことになる．そのため強化学習で用いる学習空間も細分化を行うたびに大きくなるように変更している．この時新たにできた状態に対応する学習空間は通常の初期値が与えられる．つまり新たに区別可能になった状態については試行錯誤を繰り返し学習していく必要がある．例えばQ学習であれば学習空間は「状態」・「行動」・「価値」の 3 次元から構成される．認識可能な状態数が増えれば「状態」軸の項目が増え，学習空間が増える形になる．

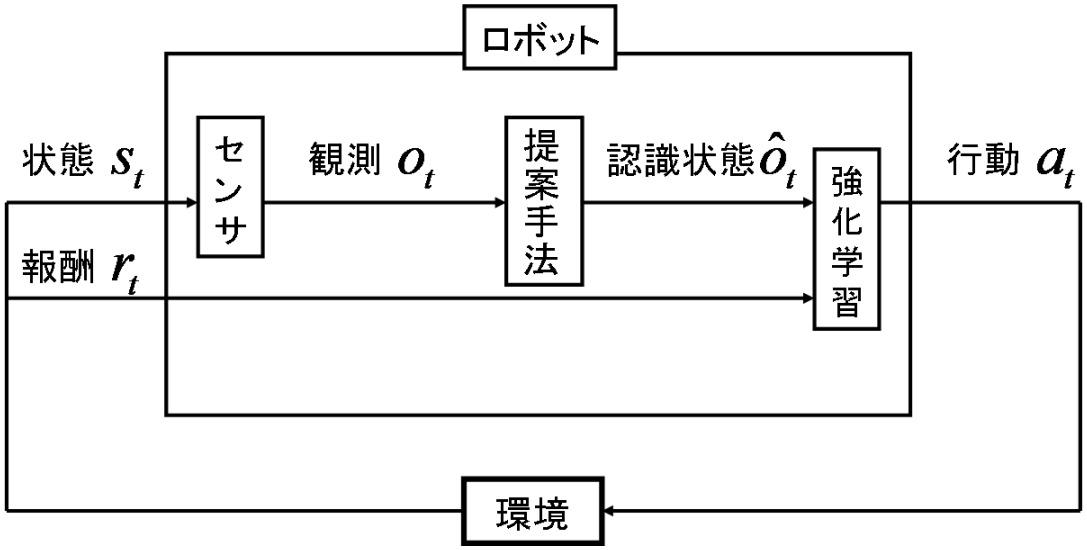


Fig. 4.12：強化学習と提案手法の関係

第 5 章　不完全知覚に対する提案手法の有効性

本章では提案手法が不完全知覚に対して有効に働くかどうかをシミュレーションを通してみていく．ここでは 4 章で提案した効果学習と組み合わせたシステムを提案システムとして利用する．また，比較対象として認識方法の異なる強化学習エージェントを 2 体用意する．1 体は不完全知覚を起こすエージェントであり，もう 1 体は不完全知覚を起こさないエージェントである．この不完全知覚を起こすエージェントに対して提案システムを適用したものを提案システムを有するエージェントとして，3 体間の学習の様子を比較していく．

5.1　実験概要

3.3 節で行われたシミュレーションと同等のシミュレーションを行っていく．迷路問題を用いて提案手法が不完全知覚に対して有効であるかどうかを検証する．その概要を Fig. 5.1 に示す．本シミュレーションでは比較対象を含めて 3 体のエージェントを用いる．1 体は不完全知覚を引き起こす可能性のあるエージェント，1 体は不完全知覚を起こさないエージェント，もう 1 体は不完全知覚を引き起こす可能性があるが提案手法を適用したエージェントである．これらのエージェントに対して同じ環境・タスクを与え，ゴールするまでの行動数に注目して結果を比較する．本実験ではエージェントがゴールするまでを 1 試行とし，エージェントがゴールしたらスタートへと自動的に戻る．そのため結果は 1 試行ごとの行動数に注目して比較していく．また，提案手法が適切に働いているかどうかを見るために，提案手法を有するエージェントが持つ状態知識の数（つまり観測を細分化した数に当たる）を学習の進行とあわせてみていく．

この実験は不完全知覚が起きるかどうかなどの環境・タスクの種類に注目して，5 種類の実験に分けて行う．5 種類の実験間では基本的に環境（迷路そのもの）とタスク（スタートとゴールの位置）以外は共通の手法・設定を用いる．細かい設定の違いはあるが，これらは学習そのものに影響を与えないものである（例えばシミュレーション終了までの全試行数など）．

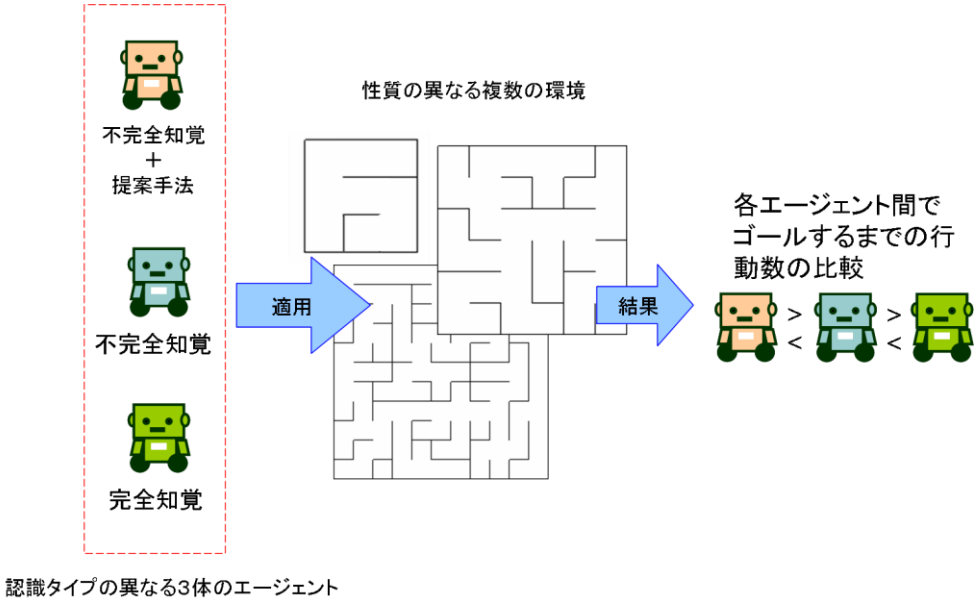


Fig. 5.1：実験概要

5.2 対象エージェント

本実験では３体のエージェントを用意する．１体は不完全知覚を起こす可能性のあるエージェントであり Agent-B とし，一体は不完全知覚を起こさないエージェントで Agent-C とする．さらに Agent-B に提案手法を適用したエージェントを Agent-A としている．

- Agent-A：不完全知覚を起こす可能性がある＋提案手法
- Agent-B：不完全知覚を起こす可能性がある
- Agent-C：不完全知覚を起こさない

Agent-A と Agent-B のモデルを Fig. 5.2 に示す．２体のエージェントは４つのセンサを所持している．各センサは全て同じセンサを用いる．センサは目の前に壁があるかないかの２種類の出力を示す．４つのセンサを用いることでエージェントは迷路において自分の周囲の壁の有無を認識できる.そのため観測の種類は Fig. 5.3 に示す 16 種類となっている．ただし Agent-A については提案手法を用いることで 16 種類以上の状態を認識することが可能となっている（あくまでセンサからの入力は 16 種類である）．Agent-A,B が選択可能な行動は４種類｛上移動，右移動，下移動，左移動｝となっており，この４つの中から１つを選択し実行する．

同様に Agent-C のモデルを Fig. 5.4 に示す．Agent-C は迷路における絶対座標をセンサを通じて得ることが出来る．そのため迷路問題において不完全知覚が起きるようなことは無い．Agent-C の観測の種類は迷路のマスの数（環境の状態の種類数）と同数である．また，認識（センサ）以外の設定は他のエージェントと共通である．

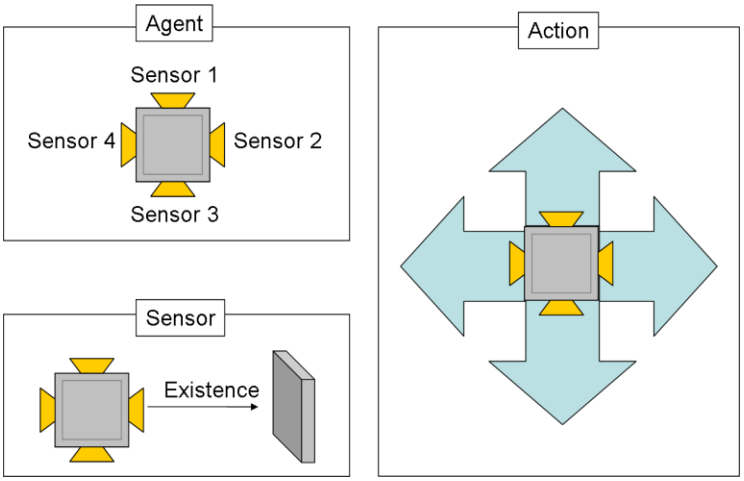


Fig. 5.2：Agent-A のモデル概要

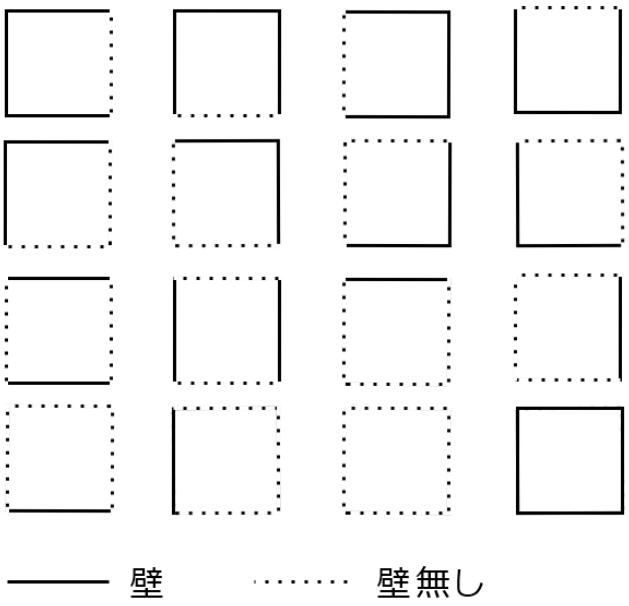


Fig. 5.3：迷路問題における Agent-A による観測の種類

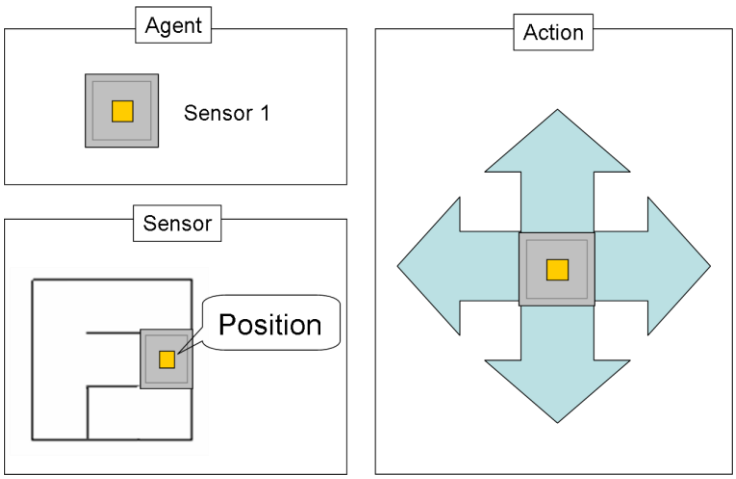


Fig. 5.4：Agent-B のモデル概要

5.3　全ての実験における共通の設定

本実験は 5 種類の実験を行うが, 全ての実験で共通している設定をここでまとめていく. 各エージェントのモデルは 5.2 節で述べたものを全ての実験で共通して用いる. そのため全ての実験で Agent-A, B, C は共通である.

各エージェントには共通の強化学習手法を適用する．今回用いたのはQ学習と呼ばれる学習手法である．Q学習では式(5.1)によって学習が進められる．さらに行動選択手法としては ε -greedy を用いる．これらの学習手法の設定は 3.3 節と同様である．これら共通のパラメータ設定を Table 5.1 と Table 5.2 にまとめる.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_{t+1}, a_t)] \quad (5.1)$$

Table 5.1：学習手法設定

学習手法	Q学習
試行数	各実験で異なる
報酬（ゴール位置のみ）	各実験で異なる
Q値の初期値	0.001
α	0.5
γ	0.7

Table 5.2：行動選択手法設定

行動選択手法	ε -greedy
ε	0.05

5.4 シミュレーション実験の種類

本実験は環境の種類・タスクの種類に合わせて 5 種類の実験を行う．各実験では環境の種類（迷路）の違い，もしくはタスク（スタート位置とゴール位置）の違いがある．それ以外の設定は 5.3 節までに述べたとおりである．以下に 5 種類の実験について挙げる．

- 実験 1 ：不完全知覚が起こらない環境
- 実験 2 ：不完全知覚が起こる環境＋タスク遂行に不完全知覚が影響を与えない場合
- 実験 3：不完全知覚が起こる環境＋タスク遂行に不完全知覚が影響を与える場合
- 実験 4：不完全知覚を起こす箇所が複数ある場合
- 実験 5：認識できる観測の種類数に対して環境の状態が圧倒的に多い場合（観測可能な数<<環境の取りえる状態数の場合）

実験 1 では Agent-A, B において不完全知覚が発生しない場合の環境を用いて実験を行う．ここでは提案手法が適切に働き，状態の細分化が行われていないことを見る．

実験 2 と 3 は 3.3 節と同じ実験を提案手法を有するエージェントを含めて再度実験を行う．この実験では不完全知覚を引き起こすポイントは 1 種類しかない．ここでは提案手法で不完全知覚に対して状態を細分化しているかどうかを見ていく．この時タスクに対して不完全知覚が影響を与えるかどうかの違いでどれだけ細分化に影響を与えるのかも見ていく．

実験 4 は本論文でのメインの実験となる. 実験 2,3 と異なり不完全知覚が起きるポイントが複数種類存在するような場合である．さらに，迷路内のほとんどのマスにおいて不完全知覚を起こすような環境を用意している．このような場合に適切に提案手法が働くかどうかを見ていく．また，状態の細分化がどの程度行われているかも合わせて見ていく．

実験 5 は今後の課題となりそうな話題を含んだ実験となっている．センサによって観測可能な数に対して環境の状態が圧倒的に多い場合，観測をいくら細分化したら良いかを見ていく．

以降は各実験を節ごとにまとめていく．以降の節ではその実験における環境設定，タスク設定，パラメータなどについて述べ，結果，考察となっている．考察は各実験ごとに行い，全体のまとめの考察としては 6 章と 7 章にて行う．

5.5　不完全知覚が起こらない環境の場合：実験 1

ここでは不完全知覚が起こらない環境を考え，その環境下での提案手法の振る舞いを見ていく．

5.5.1　環境設定

本実験で用いる迷路を Fig. 5.5 に示す．迷路の大きさは 3×3 マスである．そのためこの迷路の状態の種類は 9 種類ある. この迷路は各マスが左上のマスを原点として右方向に x，下方向に y を座標として持つ．また，この迷路において Agent-A, B は不完全知覚は起こさない．

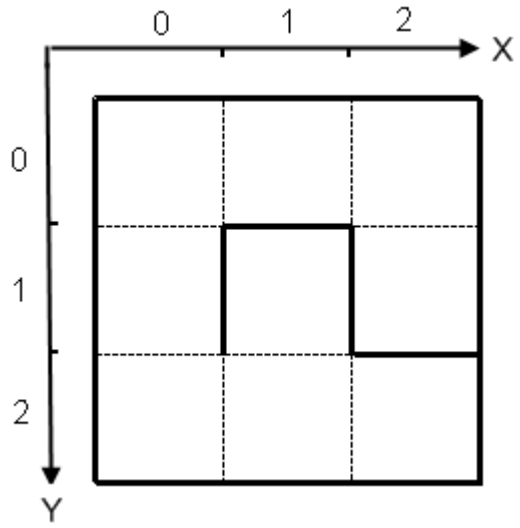


Fig. 5.5：実験環境

5.5.2　タスク設定

本実験でタスクはスタートＳとゴールＧの位置によって決まる．スタートＳは座標（0,0）とし，ゴールＧは（2,2）としている．Ｓ，Ｇを迷路にプロットした図を Fig. 5.6 に載せる．

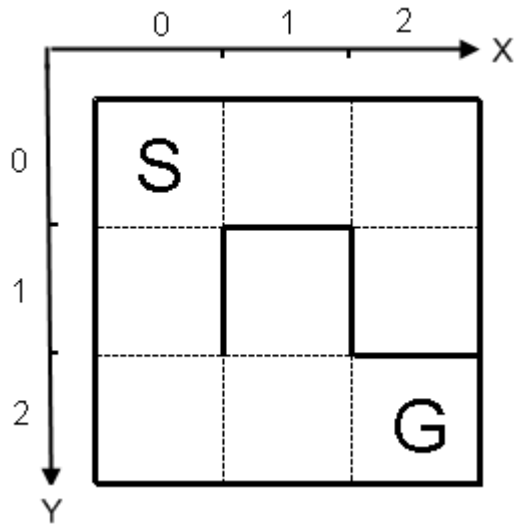


Fig. 5.6：スタートとゴール位置の関係

5.5.3　実験 1 固有のパラメータ設定

実験 1 固有のパラメータ設定を Table 5.3 と Table 5.4 に載せる．Table 5.3 は環境に関する設定，Table 5.4 は学習手法に関する設定である．

Table 5.3：環境設定

状態数	9
スタート位置座標	(0,0)
ゴール位置座標	(2,2)

Table 5.4：学習手法に関する設定

試行数	30
報酬（ゴール位置のみ）	100

5.5.4　結果

これらの設定を基に実験を行った結果を Fig. 5.7 と Fig. 5.8 に示す．まず，Fig. 5.7 は各試行ごとに見る各エージェントがゴールするまでの行動数である．5.2 節で述べたように Agent-A, B, C はそれぞれ提案システムを有するエージェント，不完全知覚を起こす可能性のあるエージェント，不完全知覚を起こさないエージェントである．X 軸に試行数を取ってあり，Trial で表されている．Y 軸はゴールするまでにかかった行動数（1 行動を 1 として数える）を表しており，Action で表される．このグラフにおいて下側であればあるほど

ゴールまでにかかった行動数が少ないことを表している．つまりグラフが右肩下がりであれば学習がスムーズに行われていると捉えることが出来る．

Fig. 5.8 は提案手法によって作られる状態知識の数を試行数毎に見たグラフである．X 軸に試行数を取り，Y 軸に状態知識の数を取っている．

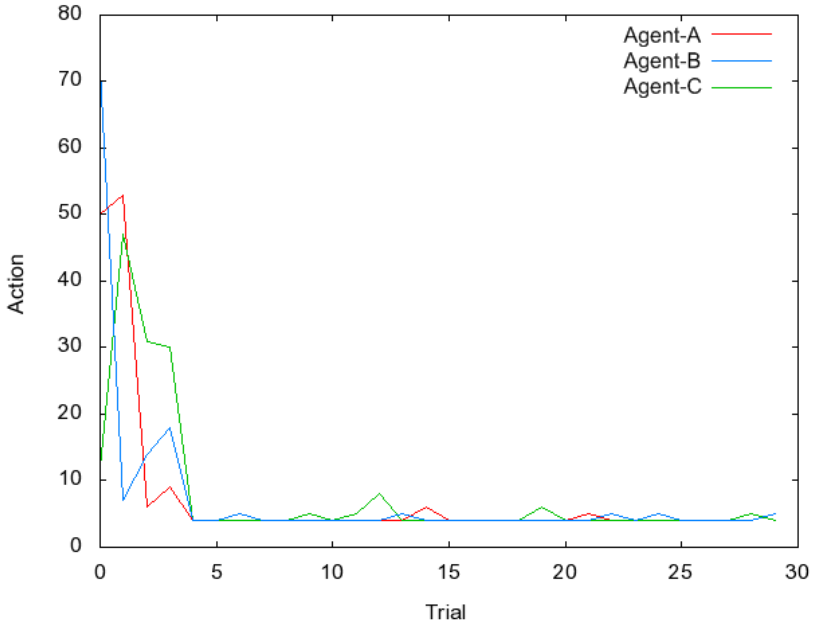


Fig. 5.7：3 体のエージェント間での各試行における行動数の比較

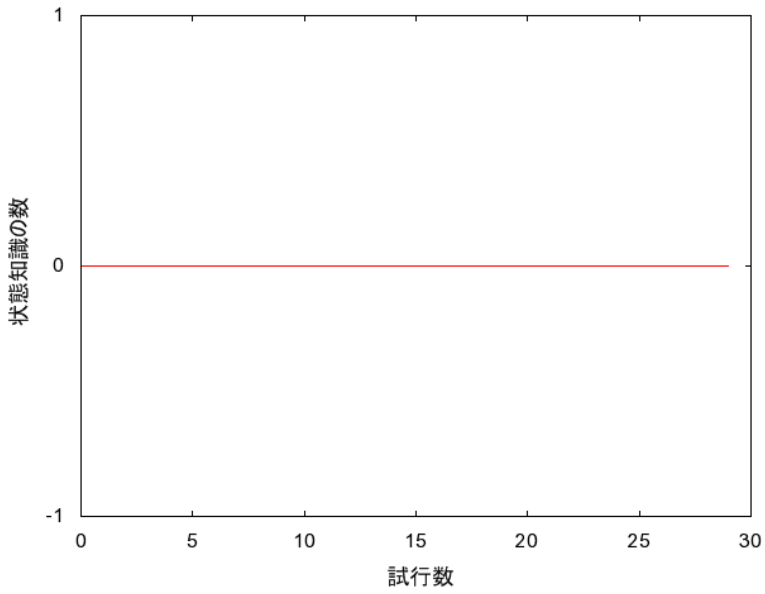


Fig.5.8：提案手法による各試行における状態知識の数の遷移

5.5.5　考察

まず，Fig. 5.7 に注目する．このグラフから最初の数試行（3 ないしは 4 試行あたりまで）まではそれぞれのエージェント間でばらつきがあるように見える．しかし，学習が収束したと見られるポイント（試行数 4 なしは 5 あたり）以降では全てのエージェントで同じような傾向が見られる．これは不完全知覚を起こさない Agent-C を中心に見ると，どのエージェントも Agent-C と大差なく学習できていることがわかる．グラフにおいて試行数が 5 以上のところで行動数が少し増える部分があるが，これは行動選択手法に ε -greedy を使っているためである．ε -greedy では ε の確率でランダムな行動を選ぶためゴールするまでの行動数が ε の確率に比例して増える可能性がある．このことは学習が進んでいようといなかろうと起こることである．つまりこの結果から不完全知覚が生じない場合はセンサの能力が不十分でも学習には支障をきたさないことが言える．

次に Fig. 5.8 に注目する．この結果は各試行における状態知識の数を示している．今回は不完全知覚が生じないため，提案手法では細分化が行われず，状態知識は増えないまま実験を終了するはずである．Fig. 5.8 の結果を見ると最後まで状態知識の数は 0 となっていることが分かるので提案手法で細分化が行われなかったことがわかる．よって提案手法で不完全知覚であると判断された観測が無かったことになる．今回の場合はこの結果から提案手法が無駄な細分化を行わないということが言える．

5.6　不完全知覚が起きるがタスクの遂行へ影響が無い場合：実験 2

ここでは不完全知覚が起こる環境を考える．また，その環境下で不完全知覚がタスク遂行に影響を与えない場合について実験を行う．この実験は 3.3 節のタスク 1 と同様の設定を用いるが，ここでも全ての設定について再度記述する．

5.6.1　環境設定

本実験で用いる迷路を Fig. 5.9 に示す．実線が壁である．迷路の大きさは 3×3 マスである．そのためこの迷路の状態の種類は 9 種類ある．この迷路は各マスが左上のマスを原点として右方向に x，下方向に y を座標として持つ．また，この迷路において不完全知覚を引き起こすマスを示した図を Fig. 5.10 に示す．Agent-A, B において同じ色（橙色）で示されたマスは同じ観測として得られる．この迷路では不完全知覚を起こすマスは座標で言うと（1,0）と（1,1）の 2 か所の一種類ある．つまり Agent-A, B は（1,0）と（1,1）のどちらのマスにいるか区別がつかないということである．ただし，Agent-A は提案手法を用いることでどちらのマスにいるか区別できる可能性がある．

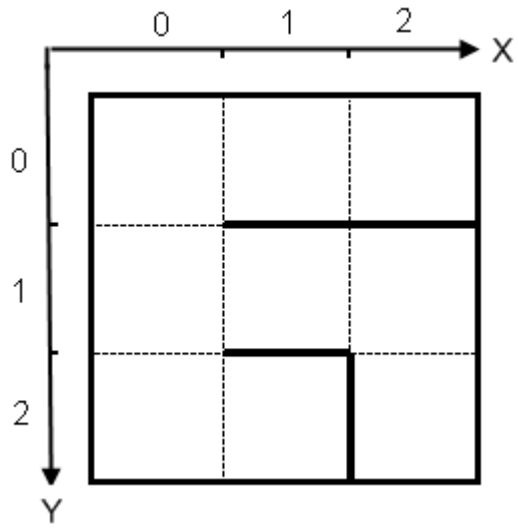


Fig. 5.9：実験で用いた迷路

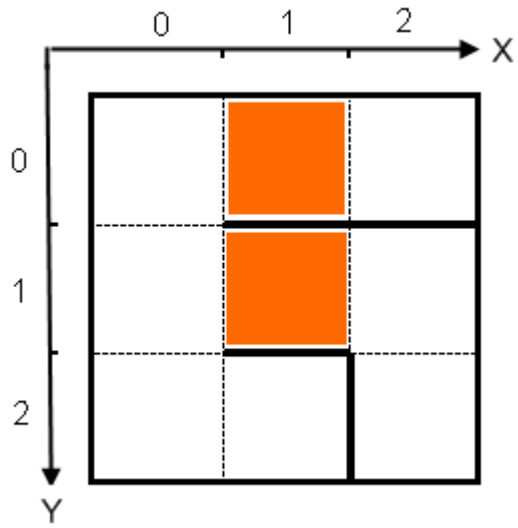


Fig. 5.10：不完全知覚を起こす箇所

5.6.2　タスク設定

本実験でタスクはスタートＳとゴールＧの位置によって決まる．スタートＳは座標（2,0）とし，ゴールＧは（0,2）としている．Ｓ，Ｇを迷路にプロットした図を Fig. 5.11 に載せる．

この設定の場合不完全知覚を起こす 2 箇所（1,0），（1,1）での最適な行動は「左へ移動」となっているのでこの二つの状態が区別できなくても学習自体はうまくいく．それについては 3.3 節で述べたとおりである．そこに Agent-A を追加しどのような振る舞いをするかを見ていく．

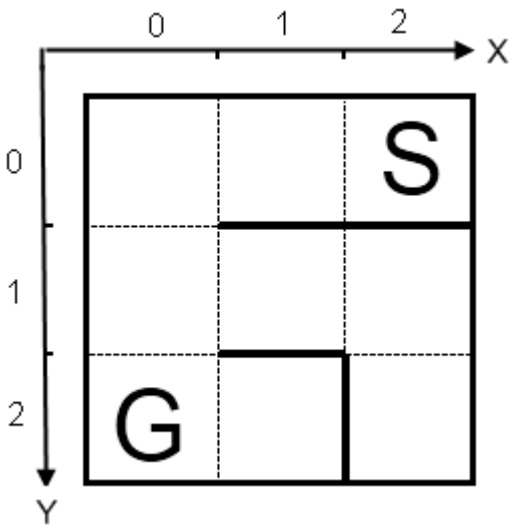


Fig. 5.11：スタートとゴール位置

5.6.3　実験 2 固有のパラメータ設定

実験 2 固有のパラメータ設定を Table 5.5 と Table 5.6 に載せる. Table 5.5 は環境に関する設定，Table 5.6 は学習手法に関する設定である.

Table 5.5：環境設定

状態数	9
スタート位置座標	(2,0)
ゴール位置座標	(0,2)

Table 5.6：学習手法に関する設定

試行数	30
報酬（ゴール位置のみ）	100

5.6.4　結果

これらの設定を基に実験を行った結果を Fig. 5.12 と Fig. 5.13 に示す. グラフは Fig. 5.7 と Fig. 5.8 と同じ形式である. ただしＹ軸のスケールは異なっている. また新たに Fig. 5.14 を載せる. このグラフは Fig. 5.13 の試行数ごとを行動数ごとに変えてみたものである. Ｘ軸に行動数，Ｙ軸に状態知識の数を取っている.

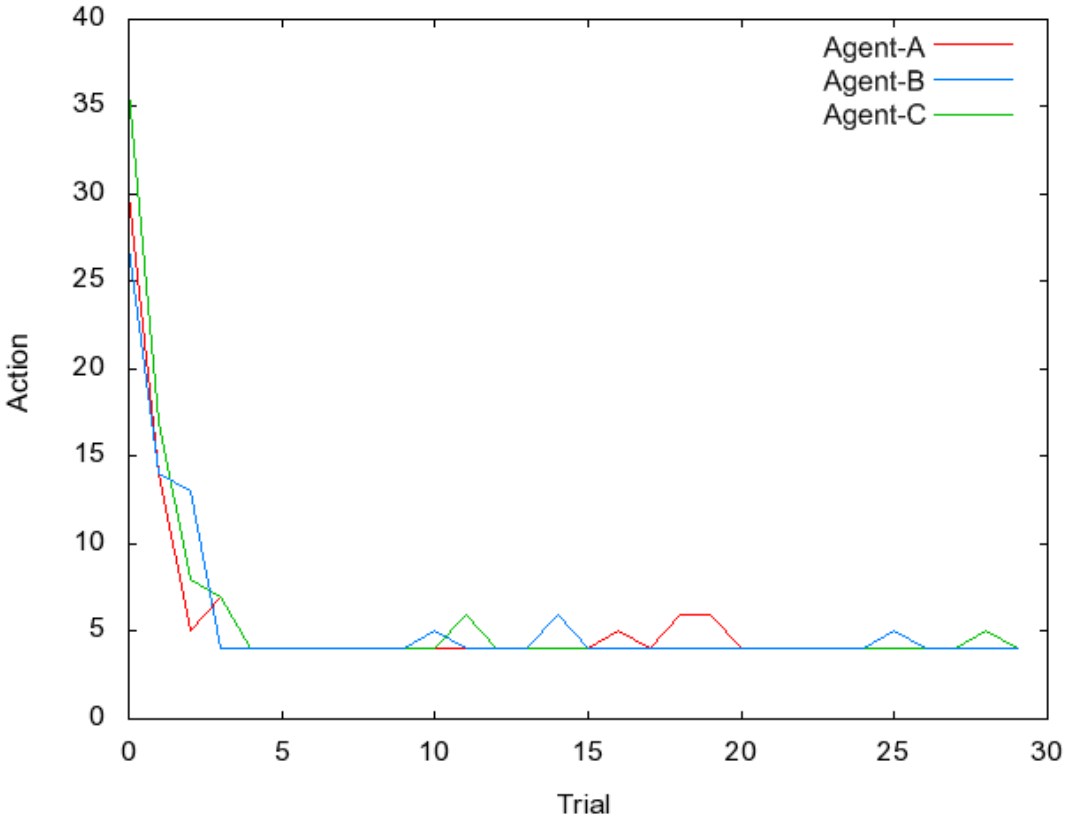


Fig. 5.12：3 体のエージェント間での各試行における行動数の比較

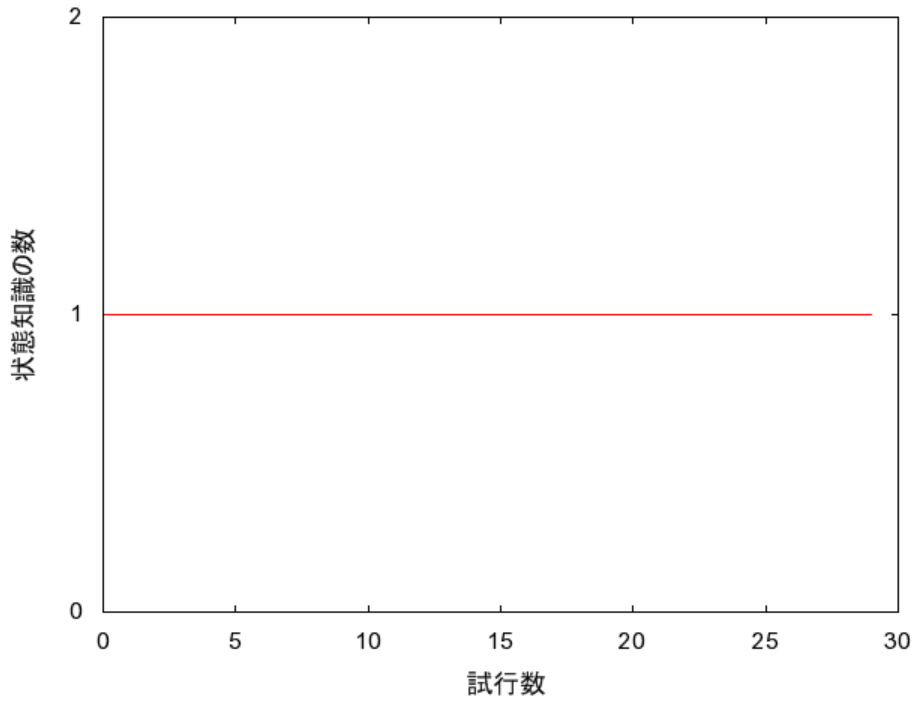


Fig.5.13：提案手法による各試行における状態知識の数の遷移

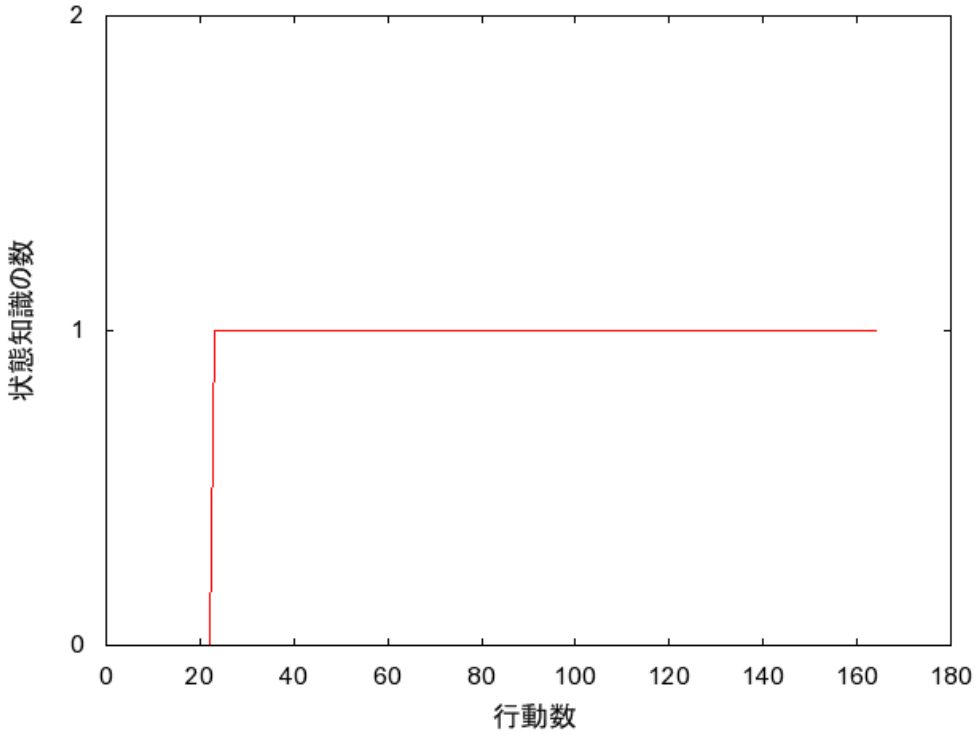


Fig.5.14：提案手法による各試行における状態知識の数の遷移

5.6.5　考察

Fig. 5.12 に注目すると実験 1 と同じ結果が得られた. 今回の環境とタスクの設定の場合, 不完全知覚であったとしても学習が進むことは 3.3 節で実証済みであった. Agent-A も例外ではなく学習が進んでいることがわかる．この時の状態知識の数に注目（Fig. 5.13）すると 1 試行目の時点で状態知識が一つ得られている．これは細分化が 1 度だけ行われたことを示している．実験 1 とは異なり不完全知覚となるポイントがあるためこうした結果が生まれたと考えられる. 1 試行目のどの辺りで細分化が行われたのかは Fig. 5.14 から分かる. おおよそ 21 試行目あたりで細分化が行われている．このことを Fig. 5.9, 5.10, 5.11 を踏まえて考えると，スタートから座標（1,1）までの最短行動数は 4 である．さらにその場所で行動することを考えると行動数は 5 と考えられる．各マスにおいてエージェントは 4 つの行動を選択可能であることから 4×5 で 20 行動数かかっていてもおかしくは無い．つまり過去の（1,0）での経験と（1,1）のときの行動の結果を比較するまでに 20 行動程度かかったのだろうと考えられる．結果的に提案手法は不完全知覚を認識し観測を細分化することが出来たようである.

5.7　不完全知覚がタスク遂行へ影響を与える場合：実験 3

ここでは実験 2 と同じ迷路で異なるゴールの位置（異なるタスク）を考える．この実験は 3.3 節のタスク 2 と同様の設定を用いるが, ここでは実験 2 と重なっている設定は省きそれ以外の設定について再度記述する．

5.7.1　環境設定

環境の設定については 5.6.1 項の実験 1 と同じものを用いる．そのためここでの詳細は省く．

5.7.2　タスク設定

スタートＳは座標（2,0）とし，ゴールＧは（2,2）としている．Ｓ，Ｇを迷路にプロットした図を Fig. 5.15 に載せる．

この設定の場合不完全知覚を起こす 2 箇所（1,0），(1,1）での最適な行動はそれぞれ「左へ移動」,「右へ移動」となっているのでこの二つの状態が区別できないと学習が上手くいかない．それについては 3.3 節で述べたとおりである．そこに Agent-A を追加しどのような振る舞いをするかを見ていく．

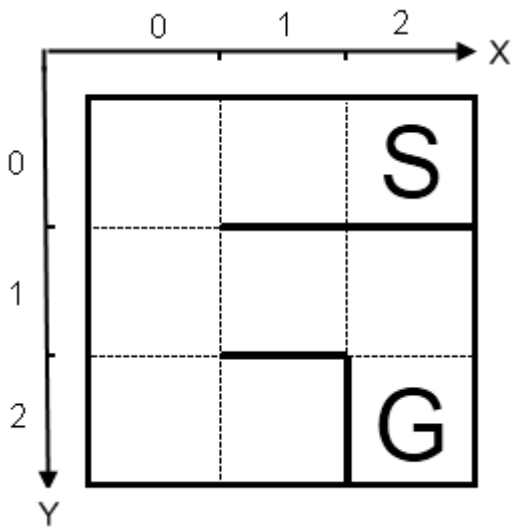


Fig. 5.15：スタートとゴール位置

5.7.3　実験 3 固有のパラメータ設定

実験 3 固有のパラメータ設定を Table 5.7 と Table 5.8 に載せる．Table 5.7 は環境に関する設定，Table 5.8 は学習手法に関する設定である．

Table 5.7：環境設定

状態数	9
スタート位置座標	(2,0)
ゴール位置座標	(2,2)

Table 5.8：学習手法に関する設定

試行数	30
報酬（ゴール位置のみ）	100

5.7.4　結果

これらの設定を基に実験を行った結果を実験 2 と同様に Fig. 5.16 と Fig. 5.17, Fig. 5.18 に示す.

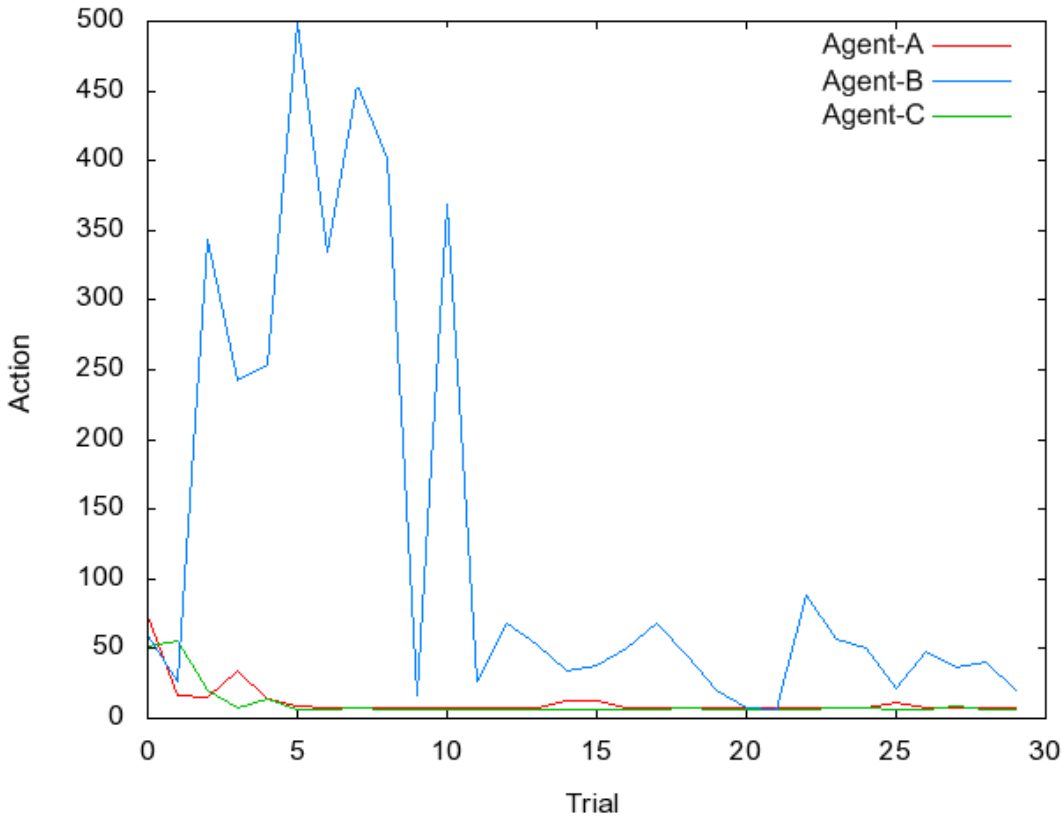


Fig. 5.16：3 体のエージェント間での各試行における行動数の比較

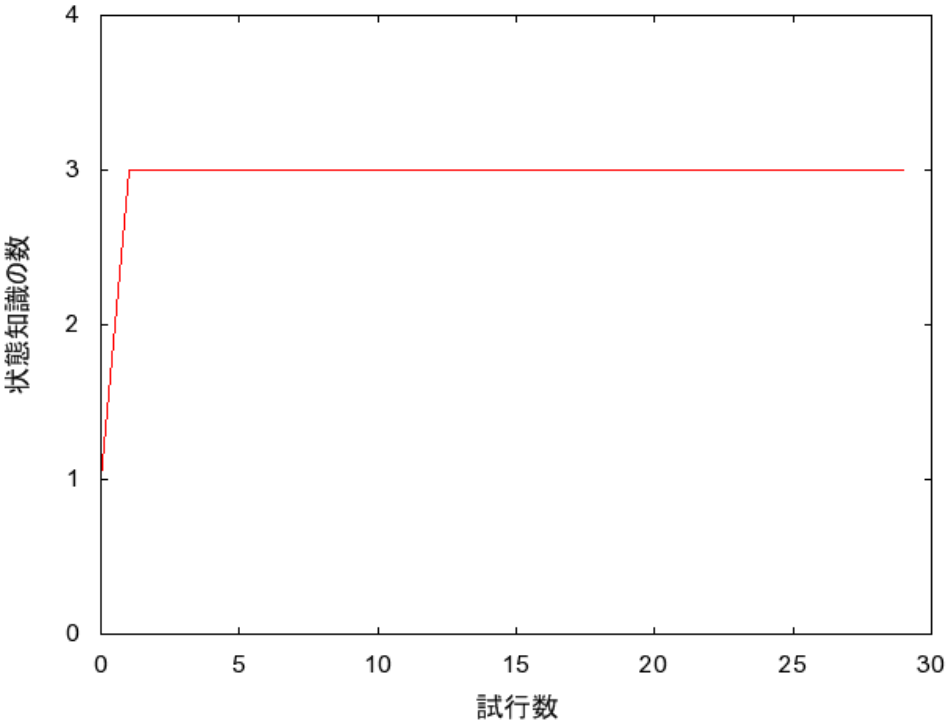


Fig.5.17：提案手法による各試行における状態知識の数の遷移

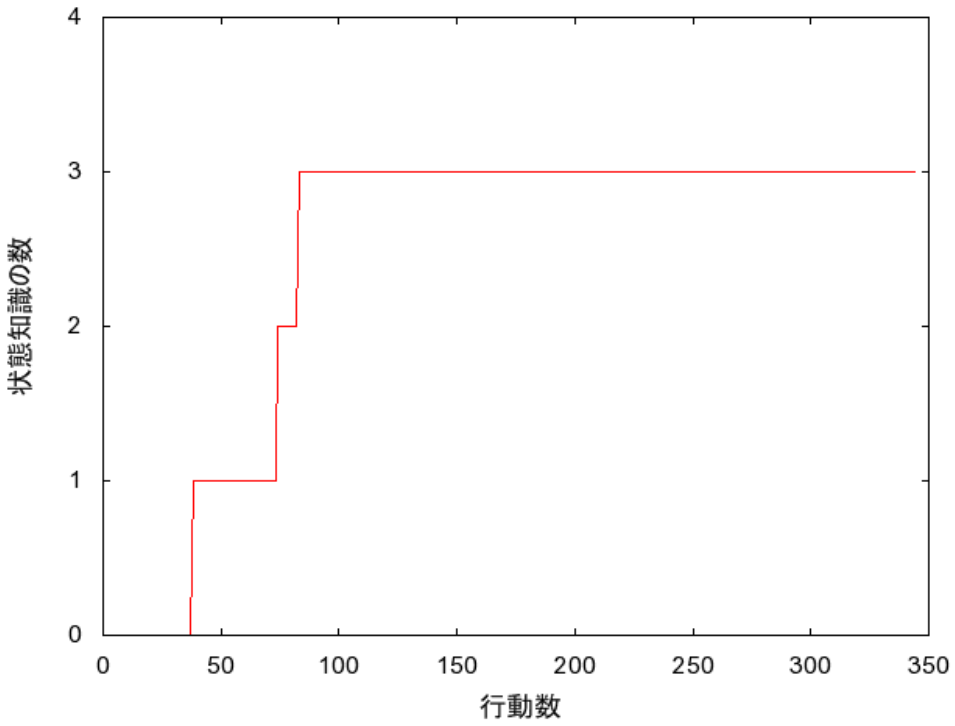


Fig.5.18：提案手法による各試行における状態知識の数の遷移

5.7.5　考察

Fig. 5.16 に注目すると Agent-B が学習出来ていないのに対して，Agent-A は学習が適切に進んでいることが見てわかる．さらに Agent-A のグラフの推移はほとんど Agent-C と同程度見ることができる．つまり提案手法によってセンサのみでは区別できない状態を区別することが出来たと考えられる．

Fig. 5.17 と Fig. 5.18 に注目してみると，実験 2 と同様に 1 試行目で状態知識を 1 つ獲得している．今回は４０試行付近と実験２よりも遅い段階で細分化が行われている．これは Agent-A が最初の試行でスムーズに迷路を進めなかったためである．また，実験２と異なって，その後の試行でも細分化が行われている．この理由は，細分化の手法は直前の観測と行動を用いているためである．不完全知覚が起こる座標（1,0），（1,1）ではそれぞれ両隣の観測が別々のものとして捉える事が出来るため，細分化が複数回行われる結果となった．例えば座標（1,0）は座標（0,0）から右移動してきた場合と座標（2,0）から左移動してきた場合の２つが考えられる．同様のことが座標（1,1）でも言える．さらに今回は不完全知覚の状態を区別できないとスムーズに学習が出来なく，迷路内を探索することが増える．探索が増えると必然的に過去の経験知識も増えていくため細分化される可能性も高くなっていく．こうした理由によって細分化が実験 2 よりも多く行われたと考えられる．つまり実験２では学習がすぐに収束してしまって探索の機会が少なかったために細分化の回数が少なかったのだろうと考えられる．

5.8　環境のほとんどで不完全知覚が起きる場合：実験 4

ここでは不完全知覚が複数種類起こる環境を考える．実験 2,3 では不完全知覚が 1 種類（２か所で同じ観測として得られた）だったが，本実験では複数個所に対して同じ観測として得られる場合や，異なる不完全知覚が起きる場合を総合して考える．

5.8.1　環境設定

本実験で用いる迷路を Fig. 5.19 に示す．迷路の大きさは 5×5 マスである．そのためこの迷路の状態の種類は 25 種類ある．この迷路は各マスが左上のマスを原点として右方向に x，下方向に y を座標として持つ．また，この迷路において不完全知覚を引き起こすマスを示した図を Fig. 5.20 に示す．Agent-A, B において同じ色で示されたマスは同じ観測として得られる．ただし，Agent-A は提案手法を用いることでどちらのマスにいるか区別できる可能性がある．また，異なる色は異なる不完全知覚を表している．そのため本迷路では８種類（■■■■■■■■■）の不完全知覚が存在している．また，Agent-A やＢにおいてセンサのみで得られる観測のうち，迷路内のどのマスかを一意に特定できるマスはゴールを除き 6 マスである（ゴール位置に着た瞬間にスタート位置へと戻されるためゴール位置は学習や認識の対象としては特別な扱いになる）．そのため全状態 25 のうち６状態のみ位

置的に決めることが出来, これは迷路全体の 24%に当たる. つまり環境の全状態のうち 24%のみ完全に認識できる場合がこの迷路になる.

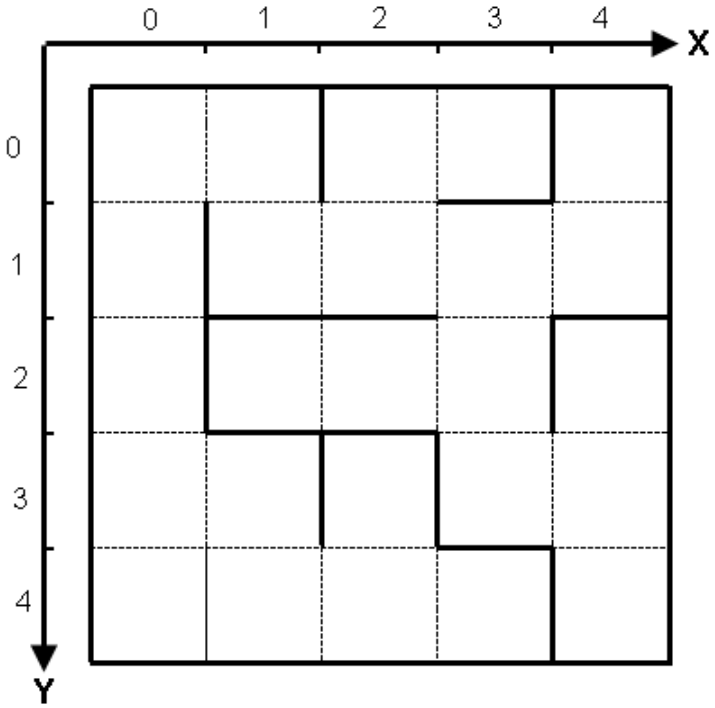


Fig. 5.19：実験で用いた迷路

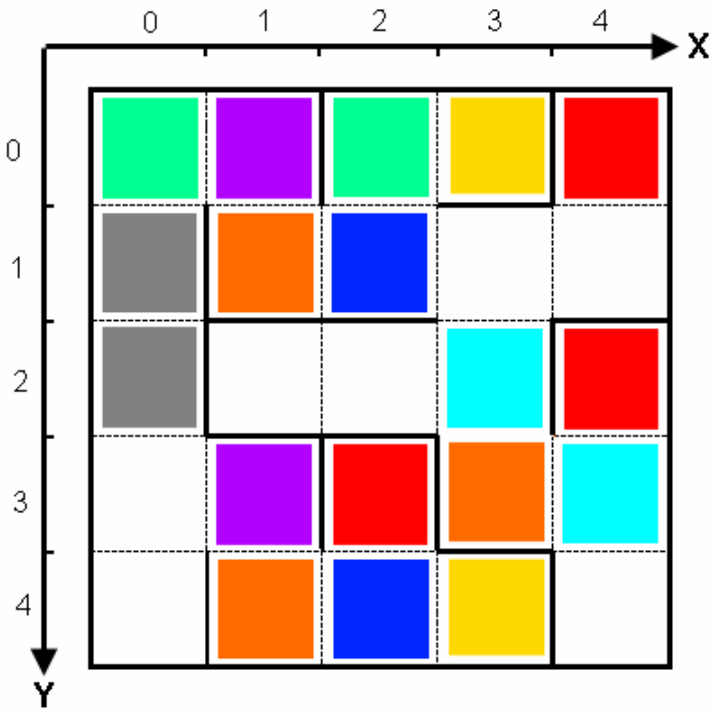


Fig. 5.20：不完全知覚を起こす箇所

5.8.2　タスク設定

スタートＳは座標（3,4）とし，ゴールＧは（4,4）としている．Ｓ，Ｇを迷路にプロットした図を Fig. 5.21 に載せる．この迷路では最短で 15 行動でゴールまで辿り着くことが可能である．

これまでの実験の結果を踏まえると 4 種類（■■■■）の不完全知覚が学習に影響を与える可能性が高いと見ることが出来る．その理由はこれらの状態においては最適な行動が異なる不完全知覚が起きているためである．

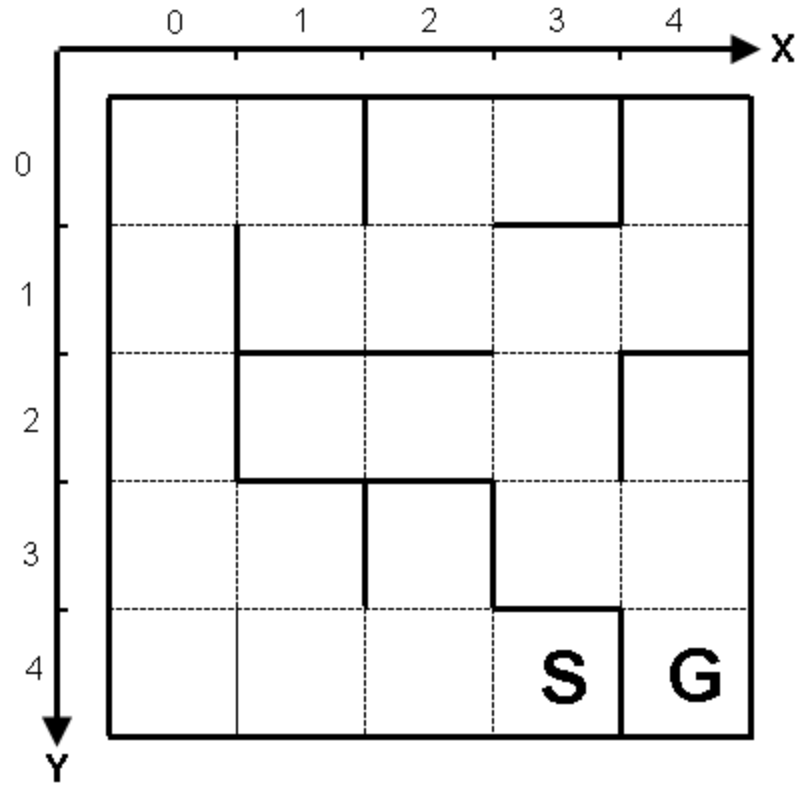


Fig. 5.21：スタートとゴール位置

5.8.3　実験 4 固有のパラメータ設定

実験 4 固有のパラメータ設定を Table 5.9 と Table 5.10 に載せる．Table 5.9 は環境に関する設定，Table 5.10 は学習手法に関する設定である．

Table 5.9：環境設定

状態数	25
スタート位置座標	(3,4)
ゴール位置座標	(4,4)

Table 5.10：学習手法に関する設定

試行数	100
報酬（ゴール位置のみ）	500

5.8.4　結果

これらの設定を基に実験を行った結果を Fig. 5.22～Fig. 5.26 に示す. Fig. 5.22, Fig. 5.23, Fig. 5.24 はこれまでと同じ形式の結果である．Fig. 5.25 は Fig.5.22 をＹ軸について拡大したものである．また，Fig. 5.26 は８種類の不完全知覚に対してそれぞれどれだけ細分化が行われたのかを表した図になっている．

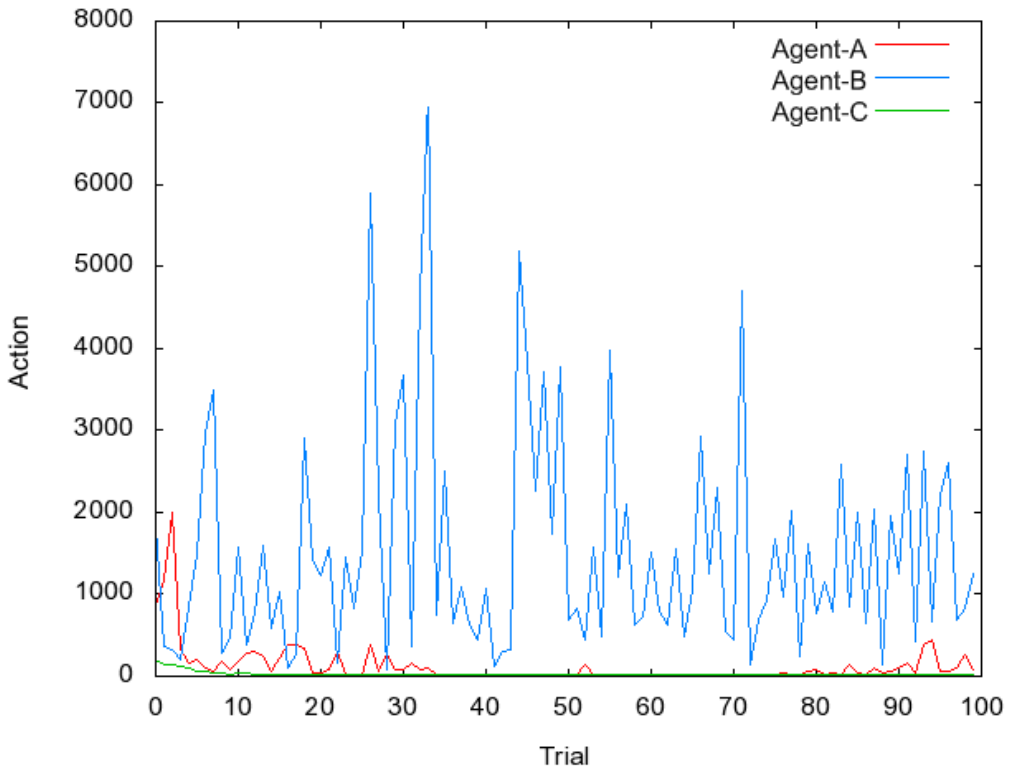


Fig. 5.22：3 体のエージェント間での各試行における行動数の比較

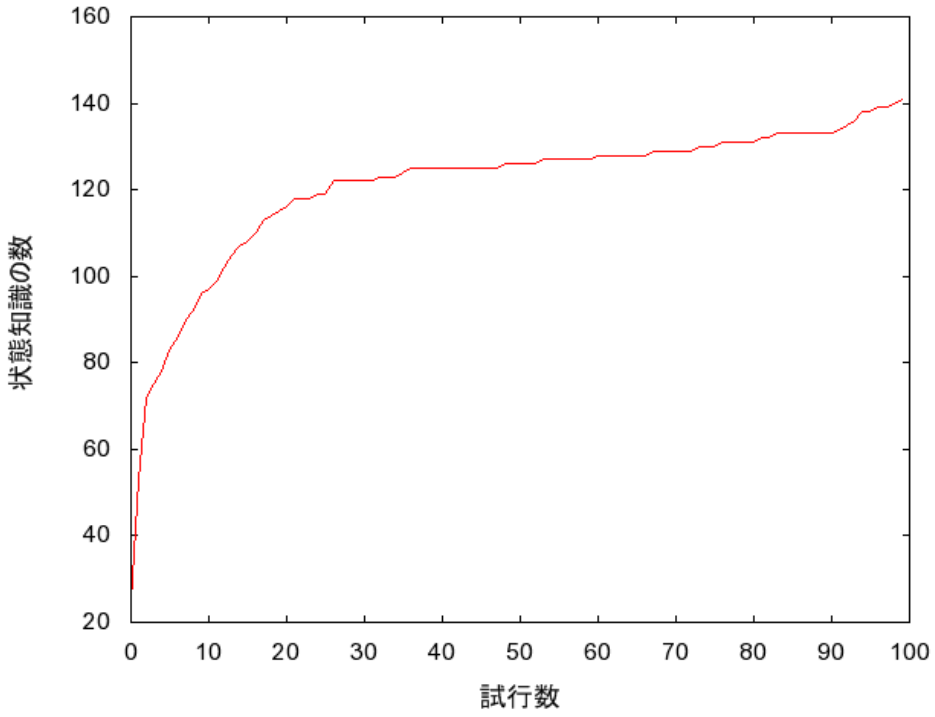


Fig.5.23：提案手法による各試行における状態知識の数の遷移

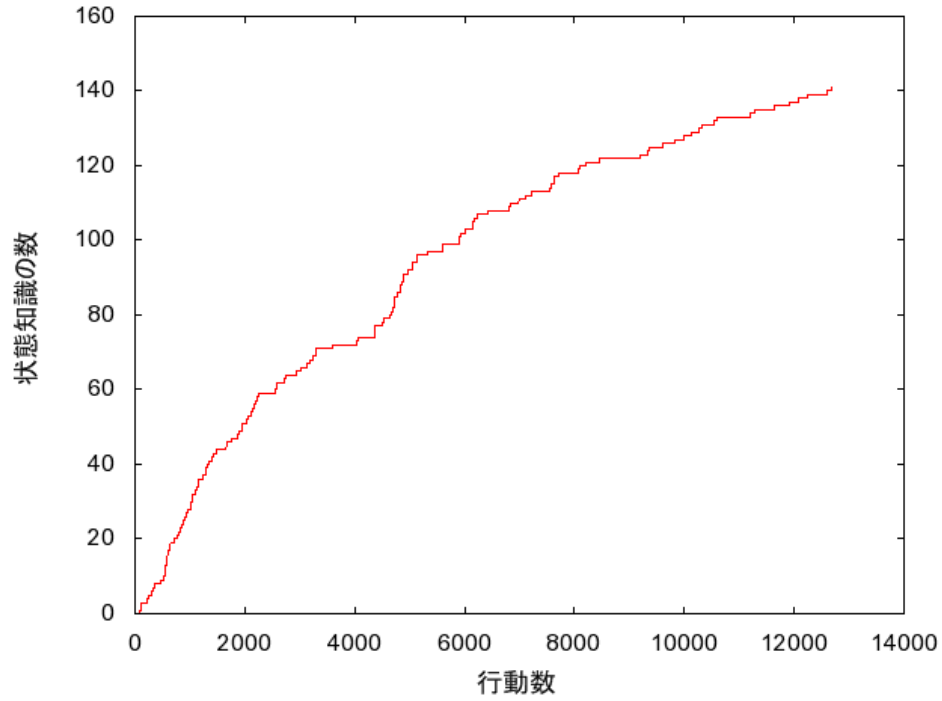


Fig.5.24：提案手法による各試行における状態知識の数の遷移

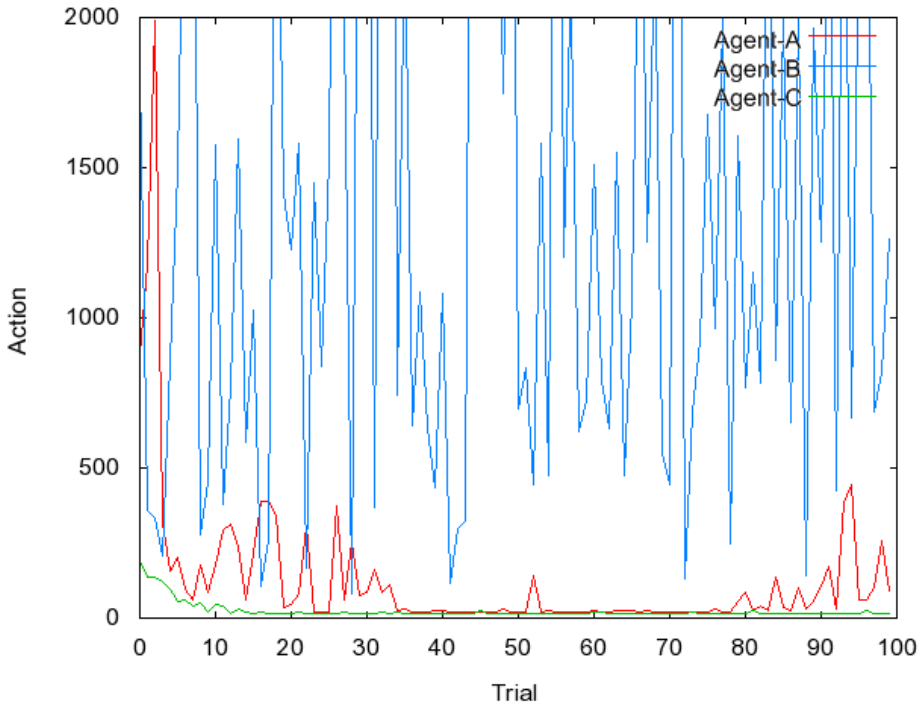


Fig. 5.25：Fig. 5.22 をＹ軸について拡大したグラフ

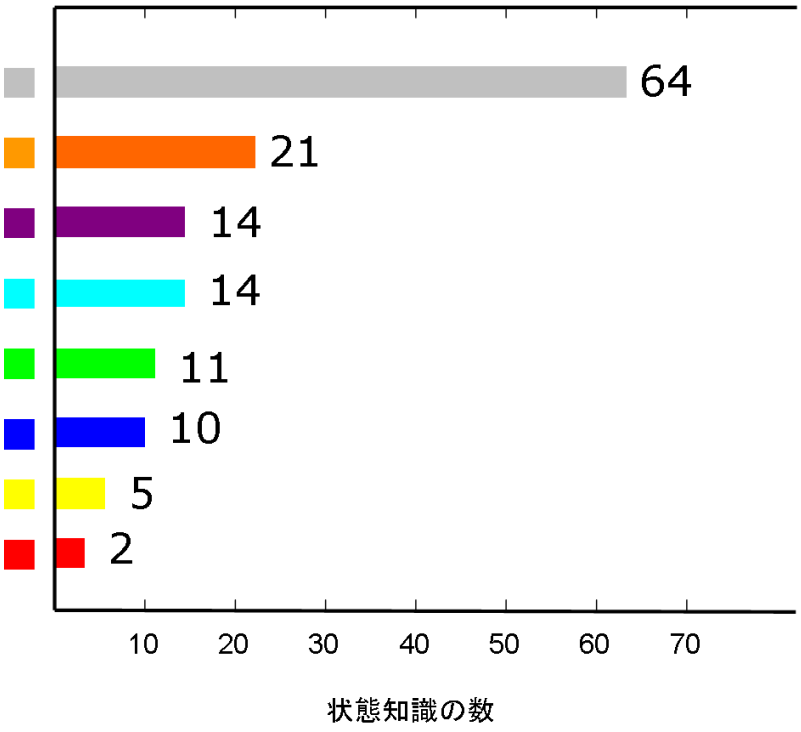


Fig. 5.26：各不完全知覚ポイントに対して作られた状態知識の数

5.8.5 考察

まず Fig. 5.22 に注目すると，Agent-A は学習の初期段階（試行数 30 以下あたり）こそ Agent-B に似た傾向を示すが，それ以降は Agent-C に近い傾向となる．つまり学習初期段階ではほとんどの観測を十分に細分化できていないために Agent-B に近い傾向が表れていると考えられる．また，学習が進むにつれ，経験知識も増えてくるため細分化が十分に行われていくため，Agent-C に近い傾向が得られたと考えられる．この理由は Fig. 5.23 を見ることで分かる．おおよそ 30 試行目付近から状態知識の数が安定し始めている．そのためタスクをこなす上である程度の細分化は出来ている状態になったと見ることが出来る．また，Fig. 5.25 を見ると，Agent-A は Agent-C に比べて安定して収束していない箇所（52 試行目周辺や 80 試行目以降など）がある．実際のところ Agent-C とまったく同じ挙動を示すわけではなく，あくまでも Agent-B より Agent-C に近い挙動となっている．おそらく細分化をしても最適解を学習するだけの細分化が行われなかったためと思われる．さらに試行数 80 試行目以降では Fig. 5.23 より状態知識が再度大きく増えている．提案手法では状態知識が増えれば学習空間も増えてしまうため，増えた分の学習にも時間が買っているのだと考えられる．

次に状態知識の数に注目する．Fig. 5.23 を見るとある程度試行が進むにつれて，安定しているようにも見える．しかし Fig. 5.24 を見ると，行動毎に少しずつ状態知識が増えていっていることが分かる．さらに Fig. 5.26 に注目する．最も多く細分化されたのは灰色（■）のマス目である．その数は本来問題となると考えられた橙色（■）の約 3 倍となっている．まずは学習にとって問題となりえる 4 種類（■■■■）のマスについて考える．この中で最も多く細分化されたのは橙色（■）である．この理由は Fig. 5.20 を見ればわかる．橙色（■）のマスの周辺を見てみると，同様に不完全知覚を起こすマスばかりである．そのため橙色（■）のマスは周りのマスの細分化が進みきらないと自身の細分化も終わることが無いのである．それに比べて紫（■）や青（■）の周辺は任意に決定できる場所があったり，大きな問題とならないような不完全知覚のポイント（■■）などがあるため橙（■）よりは少なくなったものと考えられる．水色（■）が多く細分化されているのは橙（■）の周辺にあるためであると考えられる．これらは細分化されたものがさらに細分化されることでこのように状態知識が増えているのである．

ではここで灰色（■）について考察する．このマスでは緑（■）のマスに隣接しているものの，学習に対して影響を与えるわけでもなく，周囲に任意に決定できるマスがあるにもかかわらず，状態知識の数が圧倒的に多い．この理由は簡単に言えば同じ状態が隣接しているためである．このような場合提案手法では灰色（■）のマスを延々と細分化し続けることが考えられる．この問題は今後の課題として捉える事にし，6 章にて再度詳しく取り上げて課題として記述していく．

5.9　認識可能な数に対して環境の状態数が非常に多い場合：実験 5

ここでは実験 4 の発展として，センサによる観測の種類に対して環境の状態数が非常に多い場合を考える．つまり任意に特定できる状態の割合が実験 4 の時よりも少ない場合を考える．

5.9.1　環境設定

本実験で用いる迷路を Fig. 5.27 に示す．迷路の大きさは 8×8 マスである．そのためこの迷路の状態の種類は 64 種類ある．この迷路は各マスが左上のマスを原点として右方向に x，下方向に y を座標として持つ．今回は迷路全体の約 3%が任意に特定できるマスとなっている．

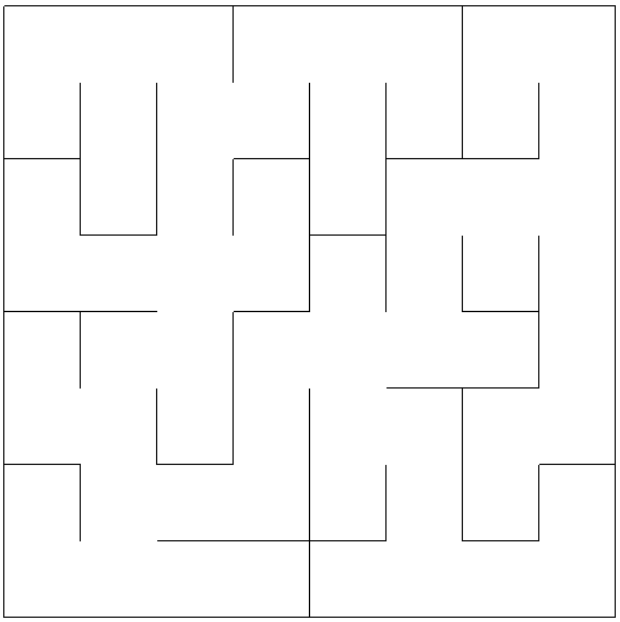


Fig. 5.27：実験で用いた迷路

5.9.2　タスク設定

スタートＳは座標（0,0）とし，ゴールＧは（7,7）としている．Ｓ，Ｇを迷路にプロットした図を Fig. 5.27 に載せる．この迷路では最短で 15 行動でゴールまで辿り着くことが可能である．

これまでの実験の結果を踏まえると 4 種類（■■■■）の不完全知覚が学習に影響を与える可能性が高いと見ることが出来る．その理由はこれらの状態においては最適な行動が異なる不完全知覚が起きているためである．

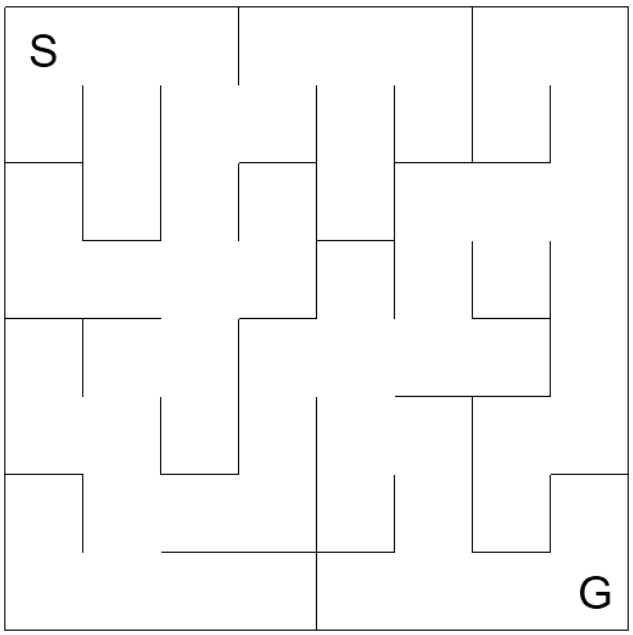


Fig. 5.21：スタートとゴール位置

5.9.3　実験 5 固有のパラメータ設定

実験 5 固有のパラメータ設定を Table 5.11 と Table 5.12 に載せる．Table 5.11 は環境に関する設定，Table 5.12 は学習手法に関する設定である．

Table 5.11：環境設定

状態数	64
スタート位置座標	(0,0)
ゴール位置座標	(7,7)

Table 5.12：学習手法に関する設定

試行数	500
報酬（ゴール位置のみ）	500

5.9.4　結果

これらの設定を基に実験を行った結果を Fig. 5.27～Fig. 5.30 に示す.

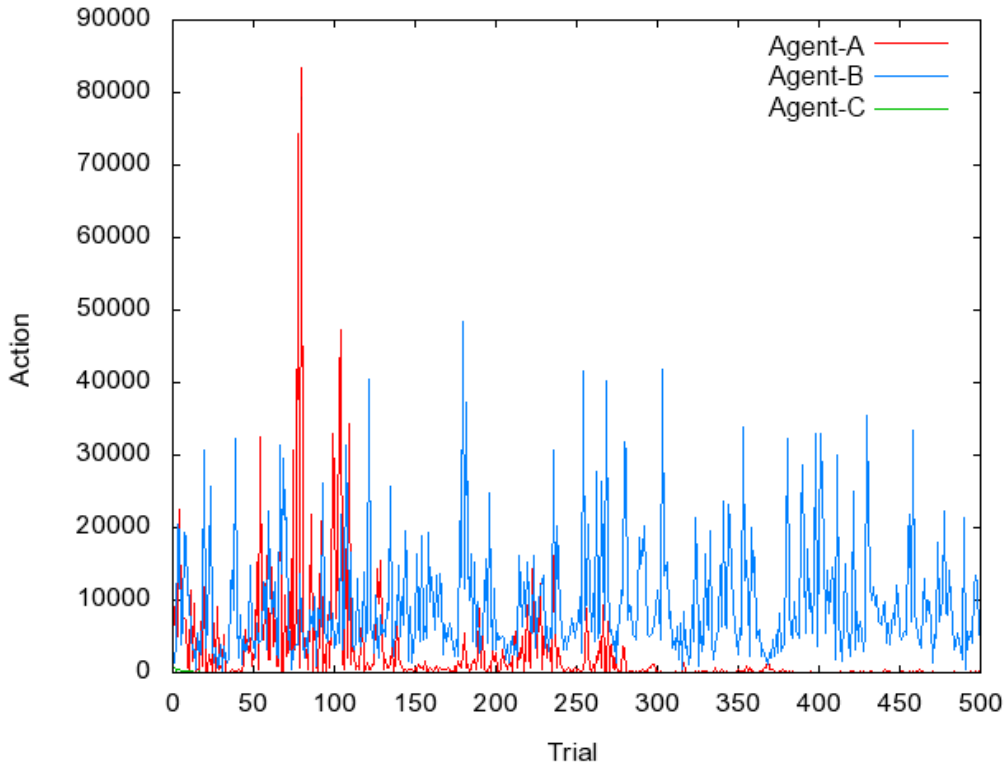


Fig. 5.27：3 体のエージェント間での各試行における行動数の比較

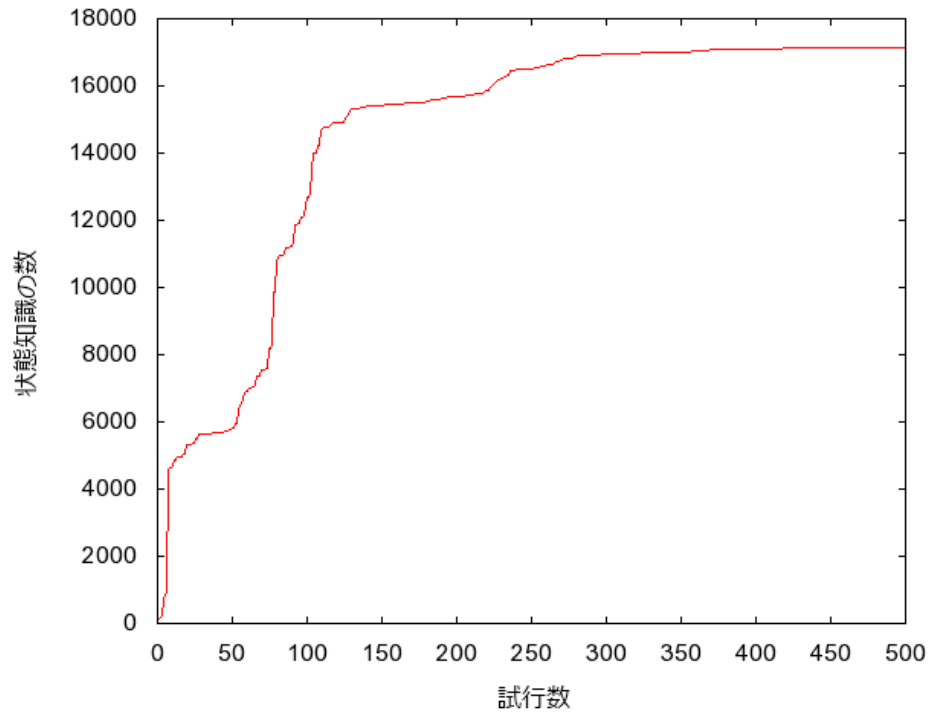


Fig.5.28：提案手法による各試行における状態知識の数の遷移

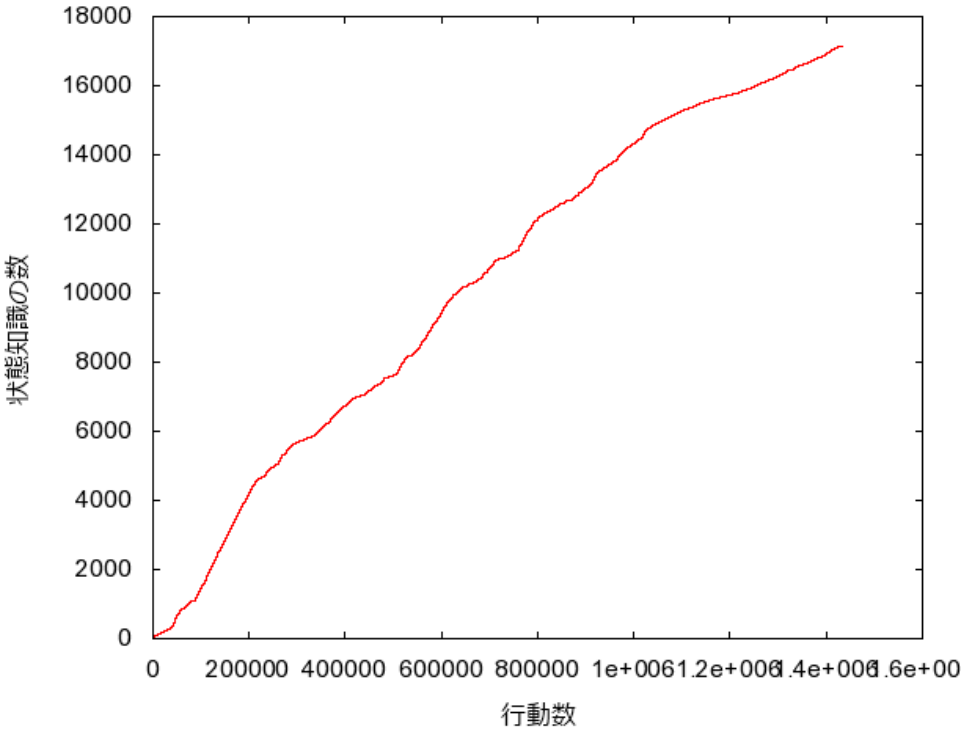


Fig.5.29：提案手法による各試行における状態知識の数の遷移

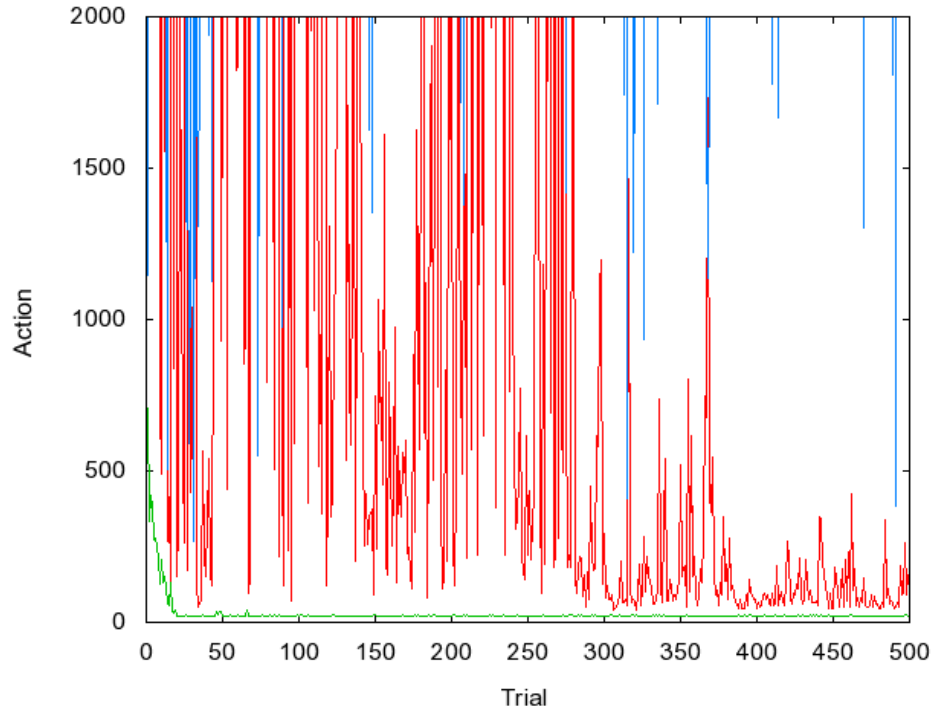


Fig. 5.30：Fig. 5.22 をＹ軸について拡大したグラフ

5.9.5 考察

この結果は実験 4 の結果に似た結果が得られた．しかし，学習初期段階において実験 4 の時よりも大きな割合で行動数が激増しているところがある（試行数 50〜100 あたり）．この時の状態知識の数をみるとやはりその数が増えていることが分かる．Fig. 5.27 において Agent-A の行動数が 50 試行以前で収束しかけているのを見ると，おそらく試行数 50 周辺から実験 4 における灰色のマスと同じような現象が起きているものと考える．つまり同じ状態に対して延々と細分化を繰り返している可能性が高い．そう考えると状態知識が増え，ゴールするまでの行動数も増えることが理解できる．しかし，こうした現象は学習初期に起きることがほとんどであり，その山を越えるとある程度ゴールするまでの行動数は安定していくことが多かった．この理由はある程度学習が進めば，探索の機会が少なくなり，決まった行動を取ることが多くなるためであると考えられる．また，Fig. 5.30 を見ると，Agent-A はある程度収束していることは分かるが，Agent-C から比べるとその振れ幅は大きく，最適な解までは学習できずにいるようである．しかし試行数を 20000 回まで取った場合だと若干であるが最適解に近付いていることは確認した．そのため試行数を無限大に増やしてやれば収束値はより最適解まで近づいていくと考えられる．

状態知識の数に注目すると試行数からみた場合は安定してきているようにも見えるが，これはゴールするまでの行動数が少なくなるからであり，実際には Fig. 5.29 にあるように行動するごとに状態知識が少しずつ増えている．どの程度の数細分化をすれば良いのかなど，今後の課題として検討するべき結果が得られたと言える．

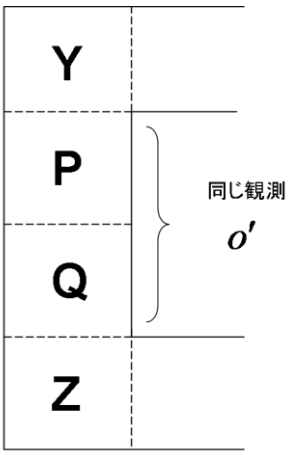
第 6 章 提案手法の問題点

6.1 迷路問題から見る提案手法の問題点

5 章の実験を通して提案手法にはある状況下では, 同じ状態に対して細分化を頻繁に行う問題が判明した．ここでは例を基にその問題について考察していく．例えば実験 4 の場合同じ状態が 2 連続で続いているとその問題が発生しているようであった．このことを具体的にロボットの行動の流れを追いながら考えていく．まず実験 4 の環境と同様に同じ状態が 2 連続で続いている環境を考える．その一部（問題となる部分を）を Fig. 6.1 に載せる．図内における Y, P, Q, Z は迷路の各マスの状態を区別したものである．この中で例として状態 Z からロボットの行動が始まったとして考える．状態 Z においてロボットが上移動した場合ロボットは経験知識として(Z, 上移動, o')が得られる．この時ロボットは状態 Q に移るが，認識は o'として認識される．同様に状態 Q において上移動を選択した場合，経験知識(o', 上移動, o')が得られる．この時のロボットの状態は P へ遷移しているが，認識は o'である．この時状態 P において上移動を選択すると，行動の結果として(o', 上移動, Y)が得られる．そうすると今回得られた結果（o'で上移動したら Y へ移った）と過去の経験（o'で上移動したら o'へ移った）の間に齟齬が生まれる．そのためこの時ロボットは新たに状態知識を作成する．この時の状態知識は x_1=(o', o', 上移動)となる．また状態知識を作成したのでこの時の経験情報としては(x_1, 上移動, Y)となる．これによってロボットは状態 Q にて上移動をしたときは特定の状態として区別することが可能になる．この時点を i とした場合のロボットが持つ経験知識 E を Table 6.1，状態知識 X を Table 6.2 に載せる．ここでロボットが再度状態 Q まで戻ってきたとする．この時先ほどと同様に上移動すると状態 P に遷移する．この時ロボットは状態知識 X によって状態 P での認識を x_1 とすることが出来，o'と区別することが出来る．この時得られる結果としては(o', 上移動, x_1)となる（o'で上移動したら x_1 へ移ったことになる）．しかし，ロボットの持つ経験知識 E 内には(o', 上移動, o')という知識がある（o'で上移動したら o'へ移ったという経験）．そのため再度 x_1 を再分割することになる．この時ロボットによって新たに定義される状態知識 x2 は x_2=(x_1, o', 上移動)となる．しかしこの x_2 は実質的に x_1 と同じであり，細分化する必要がないのである．同様にして細分化が繰り返され，意味のない状態知識が頻繁に作成されていくことが問題となっている．

つまりこの問題は細分化することで経験知識にある過去の経験と現在の認識の間に誤差が生まれてしまうことである．簡単にいえば細分化できることによって，本来は同じ状態のはずなのに違う状態として区別してしまっているのである．

この問題の原因はいくつか考えられる．経験知識の形式や細分化の方法，不完全知覚かどうかの判断方法等が考えられる．今後はこの問題を課題として提案手法のアルゴリズムの強化を図る必要があると考える. その足掛けとして 6.2 節で改善案について簡単に触れていく．



Y, P, Q, Z: 状態（各マス）

Fig. 6.1：迷路内における提案手法で問題が生じる部分

Table 6.1：時点iでの経験知識E

e_{i-2}= (Z, 上移動, o')
e_{i-1}= (o', 上移動, o')
e_i=(x_1, 上移動, Y)

Table 6.2：時点iでの状態知識X

x_1=(o', o', 上移動)

6.2 不完全知覚問題への提案手法の改善

6.1 節で話したように，この問題には様々な原因が考えられる．そこでここでは現在考えている 2 案を簡単に述べていく．これらの案を基に提案手法のアルゴリズムの強化を目指していきたい.

6.2.1 不完全知覚の判定の改善

6.1 節での例で言えば o'と x_1 を完全に区別して認識しているためにこのような問題が起きていた．本来 x_1 と o'は完全に区別するのではなく，o'の中でも特殊なやつが x_1 であるという認識であるべきである．しかし提案手法では不完全知覚かどうかを判断する場合には o'と x_1 をそのまま比較しているのでそこで問題が生じるのである．ここで o'は一度細分化されて x_1 になっているのだから，「不完全知覚ではない」と判断できればこの問題が解決できるのではないかと考えられる.

つまり現在の提案手法では不完全知覚判定部においてその時の行動の結果と経験知識を用いて判定しているが，そこに状態知識も合わせて判定することで余計な細分化を行わないようにできるのではないかと考えられる（Fig. 6.2）.

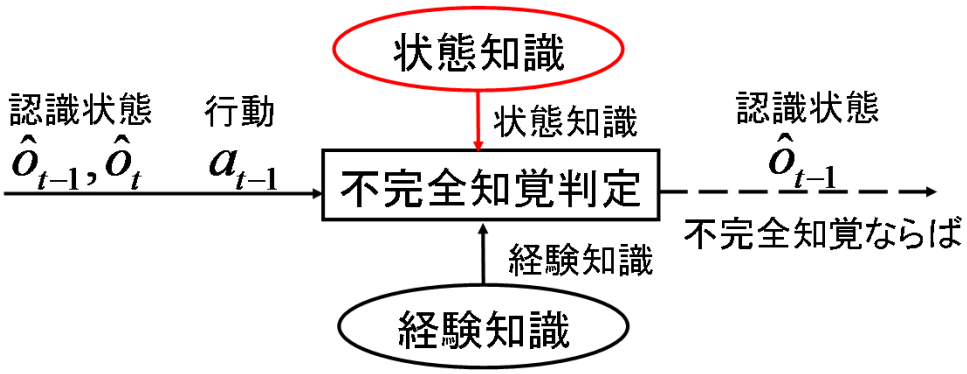


Fig. 6.2：不完全知覚判定部に注目した場合の問題点改善案概要

6.2.2　経験知識の忘却

経験知識に古い知識があるために問題が起きていると考えることも可能である．そのため細分化が行われた際に，経験知識の細分化に関わる一部を消去，もしくは適した形でのアップデートを考えることでも問題解決に繋がるはずである．さらに言えば，本実験では経験知識は無限に獲得できるものとしているが，実際にはそうはいかない．そのため経験知識の忘却については将来的に必要となってくると考えている．

しかし，この場合は経験知識を改変してしまうため，結局のところ不完全知覚の判定に影響を与える可能性が高い．そのため経験知識の忘却等，経験知識そのものに対してアプローチをかける場合はその他のモジュールとの関係も考えなくてはならない．

第 7 章 結論

7.1 まとめ

本論文では不完全知覚に注目し，不完全知覚が学習と関係が深いことを序論で述べた．特に強化学習における不完全知覚に注目し，強化学習における不完全知覚が引き起こす問題を明らかにした．3 章では実験を通して強化学習において不完全知覚が学習へ与える影響を見た．この時全ての不完全知覚が学習に対して悪影響を与えるわけではないことが判明した．実際には不完全知覚が起きていても問題なく学習が進む場合があり，それは環境やタスクによって変わってくるものだった．しかし同様の実験で不完全知覚が学習に悪影響を与える場合の結果も得られた．この場合学習が進まないといった影響がみられた．こうした場合もあることを考えて，我々は不完全知覚そのものを改善する方法を提案した．

提案する手法は，センサからの入力だけではなく，ロボット自身の過去の経験も用いて認識を行っていく手法であった．現在の情報だけではなくその直前の経験を扱って状態認識を行うことでより細かく状態を認識可能としている．提案手法では経験情報を扱って不完全知覚を認識し，不完全知覚である可能性の高い状態に対して細分化を行っている．この細分化によってより細かい状態認識が出来るようになっている．今回細分化を行うに当たって注目したのが経験情報である．ロボット自身の経験を知識として扱うことで不完全知覚の判定に利用することが可能となった．これにより不完全知覚を認識でき，効率よく細分化が出来る手法となった．

この提案手法を用いて不完全知覚が改善されるかどうかをシミュレーション実験を通して検証した．実験は環境のタイプやタスクによって 5 種類の実験を用意した．各実験を通して提案手法が不完全知覚に対して機能することが分かった．不完全知覚が起きないような場合には細分化を行わず，無駄な状態認識を行うことはなかった．不完全知覚が起きるような場兄は必要に応じて状態を細分化していることが確認できた．複数の不完全知覚が起きている場合でもある程度学習が出来るまでに認識を細分化できていることが見ることができた．ただこの実験を通して提案手法の課題も見つかった．ある程度環境が大きくなった場合等において同じ状態に対して細分化が頻繁に繰り返されるような現象が見られた．この問題は提案手法のアルゴリズム部分に問題があると考えられる．そのため今後は細分化する方法もしくは不完全知覚と判定する部分において改善する必要があると考えている．

また，提案手法は認識についての手法であるため強化学習以外でも扱える可能性がある．しかし，経験知識や状態知識などで扱っている情報は離散化された環境や時間で扱われているものであった．そのため，学習手法によっては連続的な空間や連続的な時間に対して有効な方法を考える必要がある．適用する学習手法に合わせてカスタマイズが必要となるだろうが，我々は提案手法はその他の学習手法へ適用できる可能性を秘めていると考えている．

7.2 今後の展開

現在の問題点としては 6 章で取り上げたような問題がある．こうした問題を解決することを今後の最初の課題としているが，その先の展開としては様々なものを考えている．ここではそのいくつかを簡単にまとめ上げていく．

7.2.1 動的環境下における検証

本論文では，提案手法の有効性を静的環境下で行った．そのため動的環境下への適応を考えている．現在の提案手法では，静的環境下に適した形での知識を有している．例えば経験知識であれば確定的に遷移することが前提で知識化を行った．そこで今後は動的環境下へも対応できるようにするために，知識の形式についても議論する余地があると考えている．

そのため，まずは動的環境下においてどの程度現在の手法が有効に機能するのかを見ていく必要があると考えている．こうした動的環境下での検証を基に提案手法の動的環境下への適応を目指したいと考えている．

7.2.2 連続的な環境・時間への対応

本論文では強化学習に注目して認識の手法を提案した．そのため今回扱った環境や時間は離散化された環境・時間を用いていた．これは強化学習では一般的ではあるが，現実的には連続的な空間を扱えた方が望ましい．強化学習自体，連続的な空間を扱うことは苦手なため今回は離散化された環境・時間に的を絞って認識手法を提案した．そのため今後の方針の一つとして連続的な環境や時間を扱った認識を行える手法を目指したいと考えている．

例えば連続的な空間をロボット自身が学習に適した形で離散化を行えるような認識方法等であれば，強化学習にも適用することが出来る認識手法となると考えられる．

7.2.3 実ロボットへの適用

この研究の最終的な目的地としては，実ロボットへの提案手法の適用を考えている．しかし，実ロボットの場合これまでに挙げた様々な課題をこなさなくては適用が難しいと考える．

例えば実ロボットの場合メモリの限界があるため，ロボットが持つ知識にも限界が生じる．そのために知識の忘却について考えなくてはならない．また実ロボットを扱う場合周りの環境は動的な環境であることが多いと考えられる．さらに現実は連続的な環境・時間のため 7.2.1 項や 7.2.2 項で挙げたことを先に考えていく必要がある．

将来的にはこうした課題を乗り越えて，実ロボットにおいて学習手法にとらわれずに不完全知覚の解決に図りたいと思っている．

付録 「強化学習」

1 強化学習概要

1.1 強化学習とは

強化学習はロボット（エージェント）などにおいて経験的に物事を学習していく学習方法となっている．ロボットは自分を含む周囲の環境を認識し，より良い行動をとるように学習を進めていく．

強化学習で扱う環境は基本的なものだと MDP としてモデル化された環境である．この環境モデルに対しては有限回の試行で最適な解を見つけ出すことが可能であることが証明されている．ただし，マルコフ決定過程としてモデル化されていない環境でも強化学習を用いることは可能であるが，この場合最適な解が見つかることは保障されていない．

強化学習では報酬と呼ばれるスカラの値を用いて学習を行う（Fig. 1）．ロボットは行動を選択するとその行動の結果に見合った報酬を受け取る．受け取った報酬を基にその行動が良かったのかどうかを判断する．この報酬は人間によって設定されるため，実際にロボット等に学習を行わせる際には目的に合わせて人間が報酬を設定してやる必要がある．

強化学習は大きく 2 つのパートに分かれている．一つは学習部，もう一つは行動選択部である．学習部と行動選択部については次の節以降で詳しく話をする．

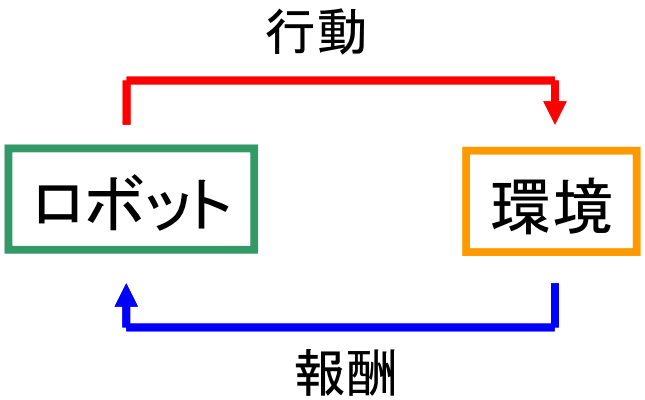


Fig. 1: 強化学習における行動と報酬の関係

1.2 強化学習の構成要素

強化学習を構成する要素としては以下のものが挙げられる．

- ロボット（エージェント）

  実際に学習を行うもの．コンピュータ上で行うシミュレーションの場合にはエージェントと呼ばれることが多い．ロボットはセンサを有し，周りの環境を認識

することが可能である．また，ロボットは何らかの行動を取ることができる．ただし取れる行動はロボットの身体構造に依存する．

- 環境

  ロボットを取り巻く周りの状態．例としてロボットが家の中に置かれているとすると環境は家の中となる．この家の中にテーブルや電気があり，これらは環境の要素である．ロボットが認識する状態はこの環境の要素の状態によって決められる．また，認識できる状態は所持しているセンサに依存する．

  ロボットの行動やそれ以外の要因で変化したりするような環境は動的環境と呼ばれる．これに対して変化が起きない環境は静的環境と呼ばれる．

- 報酬関数

  報酬関数は強化学習において目的（タスク）を表している．つまりこの関数はある状態で何が良くて何が悪いかを定義したものである．基本的にはロボットが認識したある状態に対してスカラの値が報酬として定義される．ロボットは行動を取ることで別の状態に遷移し，遷移した先の状態から報酬を貰うということになる．この報酬関数は目的に応じて人間が定義するのが一般的である．

- 学習部

  実際に学習を行う部分で，目的に対して学習が行われる．つまり受け取った報酬を基に自分のとった行動の評価が行われる部分である．この学習部では価値関数と呼ばれるものを利用する．報酬関数が即時的な意味合いで何が良いかを表しているのに対して，価値関数は最終的に何が良いかを表す．報酬関数によって学習が行われた結果が価値関数であるとも言える．よってこの価値関数はロボットが変更を行うものである．

- 行動選択部

  ここでは学習部で行われた評価の結果から次に取る行動を選択する．

1.3 強化学習の流れ

強化学習の流れは Fig. 2 のようになっている．

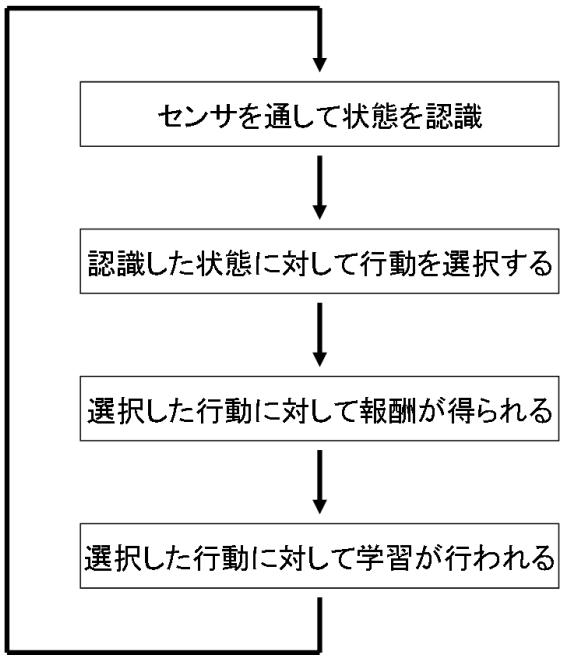


Fig. 2: 強化学習の流れ

ロボットやエージェントはセンサを有しており，このセンサで自分が置かれている状況を認識する．そしてそれまでの学習の結果から認識した状態に対して行動を選択する．この時どの行動を選択するかは行動選択部の手法に依存する．この選択した行動に対して報酬を得ることで学習を行っていく．どのように学習するかは学習部の手法に依存している．ロボットやエージェントはこのサイクルを繰り返すことで目的に対してより最適な行動を学習していく．

1.4 強化学習の利点

強化学習には以下のような利点・特徴が存在する．

- 動的な環境・未知の環境を扱うことが可能

  報酬によって学習を行うことで動的な環境や未知の環境を取り扱うことが可能になっている．これは報酬構造と環境構造が別々のものだからである．このことから強化学習は動的な環境や未知の環境下でのロボットの行動獲得などの問題に用いられることが多い．

- 学習の際に教師を必要としない

  強化学習ではどのように学習していくかという学習過程まで人間が教えてやる必要は無い（教師を必要としない）．最終的な目的を報酬という形で与えさえすれば，目的を達成するまでの過程というのは自動的に学習する流れになっている．

- 試行錯誤による探索

  強化学習では試行錯誤による行動からどのように行動すれば良いかを学習する．試行錯誤によって様々な経験をすることでより最適な行動を見つけ出すことができる仕組みになっている．

- 遅延報酬

  報酬の与え方は様々な方法があるが，強化学習では遅延報酬と呼ばれる形で報酬を与える場合がある．遅延報酬では最終的な目的の状態に対してのみ報酬を設定するなど，すべての状態に対して報酬を設定するわけではない．このためロボットは自分の取った行動の良し悪しを即座に決めることができないことがあり，この時行動の評価に遅れが生じる．この遅延報酬による問題設定はより現実的な問題に対して用いられることが多い．これは実社会においてすべての状態を定義できないことが多いためである．このような場合は最終的な目的の状態に報酬を設定し，あとはロボットの試行錯誤による学習に任せるといった形になる．

2　強化学習・学習部

ここでは強化学習における学習部の話をする．学習部では手に入れた報酬に基づいて実際に学習を行う部分である．強化学習では報酬から自分の取った行動の良し悪しを判定して学習を行っている．一言で学習と言っても，どのように行動の良し悪しを判定するか，判定した結果をどのように保持するかなどの点から様々なアルゴリズムが存在している．中でも有名な手法に Q 学習と呼ばれるものがある．ここからはこの Q 学習について詳しく話をする．

Q 学習ではロボットやエージェントは学習の結果又は途中経過を示す Q 値を持つ．この Q 値は価値関数であり，将来的にどの程度報酬を貰えるかを表している．そのため Q 値は期待報酬または行動の評価値とも呼ばれている．Q 値はロボットが認識する状態とその時に取ることができる行動の一つをペアにしたものに対して与えられることが多い（Fig. 3）．行動を決定する方法は行動選択部の手法に依存する．

この Q 学習では報酬を得られたかどうかにかかわらず 1 回の行動ごとに学習を行う．このとき式（1）を用いて学習を行う．式（1）では報酬の他に，行動した先の状態が持つ Q 値を用いて行動の良し悪しを決めている（Fig. 4）．

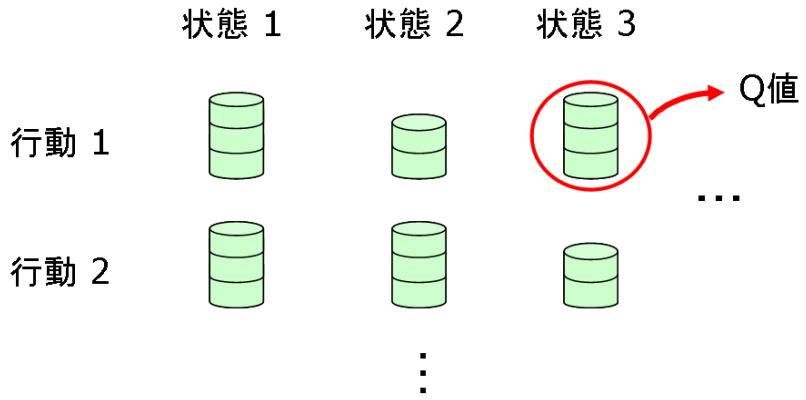


Fig. 3：Q 値の概念

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_{t+1}, a_t)] \quad (1)$$

$Q(s_t,a)$:更新対象のQ値
$\max_a Q(s_{t+1},a)$:遷移先の状態が持つ最大のQ値
s_t:遷移する前の状態
s_{t+1}:行動後の遷移先状態
a_t:行動
r_{t+1}:報酬
α:学習率($0.0 \leqq \alpha \leqq 1.0$)
γ:割引率($0.0 \leqq \gamma \leqq 1.0$)

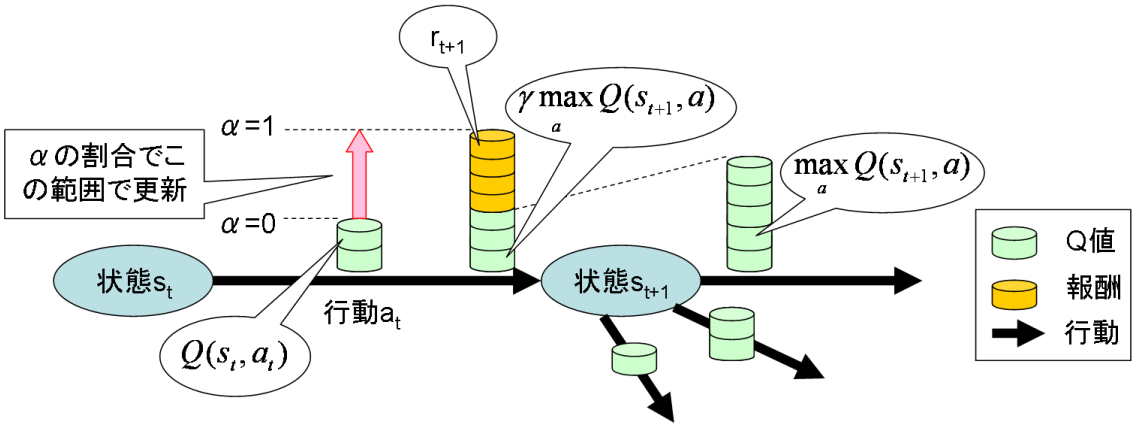


Fig. 4：Q 学習による Q 値更新

3　強化学習・行動選択部

ここでは強化学習における行動選択部の話をする．行動選択部ではロボットやエージェントが行動を選ぶ部分となっている．この時ロボットやエージェントは学習部で行っている学習の結果を基に行動を選択する．

行動選択部のアルゴリズムは，どのように行動を選択するかという点からいくつかの手法がある．ここでは greedy 法，ε-greedy 法について説明をする．

A)　greedy

greedy 手法では最も評価の高い行動を取る．つまり，ある状態で取れる行動の中で最も高い報酬が得られると思われる行動を選択する（Fig. 5）.

この手法の特徴は行動の評価に忠実に従うことである．これによってある状態において各行動の評価に差が生まれると，その状態においては行動が一意的に決まる．しかしこれでは，その他の行動がもっと良い行動である可能性については考慮しない．そのため，状態数が多い場合や行動の選択肢が多い場合には向かない.

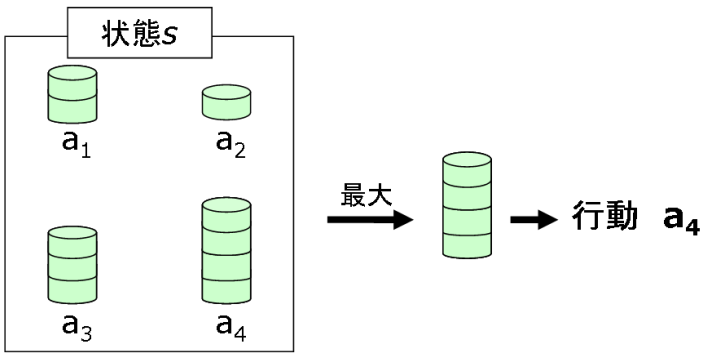


Fig. 5：greedy 法

B)　ε-greedy

ε-greedy は greedy 手法にεの確立でランダムな行動を取るようにしたものである．この時εは 0 以上 1 以下の値であるので(1-ε)の確率で greedy 法と同様に最も高い報酬が得られるだろう行動を取ることになる（Fig. 6).

この手法の特徴はεの設定を任意で決めることができるところにある．εの値を大きく設定してやれば様々な状態を経験することが可能である．これはどの行動がよりよい選択

なのかを探索するということである．また，ε を小さく設定してやれば greedy な行動を取りやすくなる．つまりより多く経験をさせて最適な行動を見つけさせるのか，少ない経験でも良いのでより早く目的を達成させるのかを ε で調節することができる．

このような特徴から行動選択の手法としてよく用いられる手法の一つとなっている．

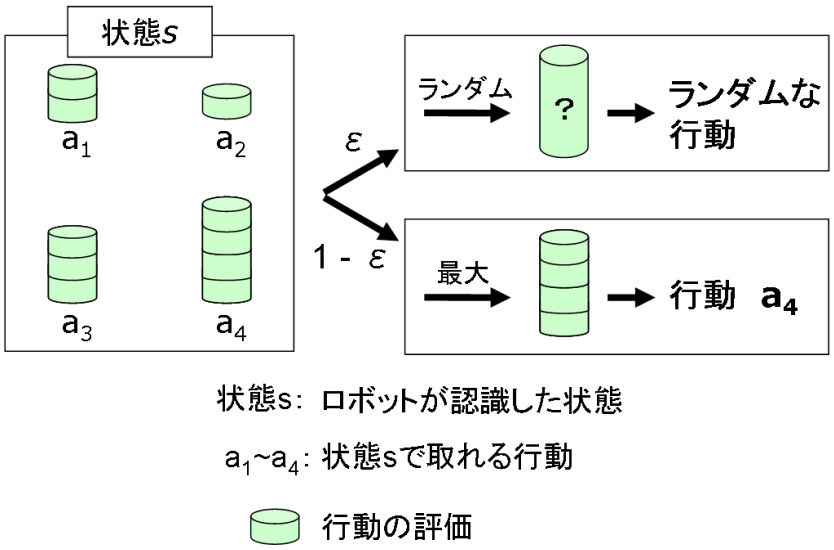


Fig. 6： ε -greedy 法

参考文献

[1] M. Hirose and K. Ogawa, "Honda humanoid robots development", Phil. Trans. R.Soc. A, Vol.364, No.1850, pp.11-19, 2007

[2] 木島康隆，“群の中の個体の知能の発達”，室蘭工業大学卒業論文，2007

[3] 木島康隆，“コミュニケーション相手の取捨選択による個体知能の効率的発達”，室蘭工業大学修士論文，2009

[4] 宮本弘之，川人光男，“作業レベルのロボット学習のための見真似による教示”，電子情報通信学会論文誌，D-II，情報・システム，II-情報処理 J81-D-2(10), pp.2401-2410，1998

[5] 北越大輔，塩谷浩之，中野良平，“BN 混合モデルを用いたオンライン型方策改善システムの動的環境への適応”，電子情報通信学会技術研究報告．NC，ニューロコンピューティング 104(349), pp. 15-20, 2004

[6] 中南義典，“環境認識能力の変化が学習に及ぼす影響について”，室蘭工業大学卒業論文，2009

[7] 木村元， Kaelbling Leslie Pack，“部分観測マルコフ決定過程下での強化学習”，人工知能学会誌，Vol.12，No.6，pp.822-830，1997

[8] 三浦純，“ロボットのための視覚環境認識”，電子情報通信学会技術研究報告. PRMU，パターン認識・メディア理解 Vol. 109，No. 88, pp. 49-54, 2009

[9] 三浦純，“移動ロボットの環境認識と行動生成”，日本ロボット学会誌，Vol.26，No.4，pp.322-325，2008

[10] 植村渉，上野敦志，辰巳昭治，“POMDPs 環境のためのエピソード強化型強化学習法”，電子情報通信学会論文誌，Vol. J88-A，No.6，pp.761-774，2005

[11] 山村忠義，馬野元秀，瀬田和久，“段階的な視覚をもつエージェントにおける強化学習について”，日本ロボット学会誌，Vol.18，No.5，pp.561-570，2006

[12] E.J. Sondik, “The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs.”, Operations Research, Vol. 26, No. 2, pp. 82-304, 1978

[13] K.J. Astrom, “Optimal control of Markov decision processes with incomplete state estimation.”, Journal of Mathematical Analysis and Applications, Vol. 10, pp. 174-205, 1965
205

[14] 藤原真一，宮本俊幸，“POMDPs 環境における状態遷移の部分的履歴を用いた強化学習手法”, 電子情報通信学会論文誌. A, 基礎・境界, Vol. J94-A, No. 10, pp. 778-781, 2011

[15] 斎藤健，増田士朗，“不完全知覚判定法を導入した Profit Sharing”，人工知能学会論文誌，AI 19，pp.379-388，2004

[16] 植村渉，上野敦志，辰巳昭治，“POMDPs 環境下での経験強化型強化学習法”，電子情報通信学会技術研究報告. AI，人工知能と知識処理，Vol. 104，No. 233，pp. 1-5，2004

[17] 齋藤宗孝，“不完全知覚問題に対する内部メモリを用いた強化学習法に関する研究”，情報処理学会研究報告. GI, [ゲーム情報学] 2004(28)，pp. 81-87，2004

[18] Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning", The MIT Press, 1998

[19] Daniel W. Ttroock, "An Introduction to Markov Process", Springer, 2005

[20] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, “Planning and Acting in Partially Observable Stochastic Domains. “, Artificial Intelligence Journal, Vol. 101，pp. 99-134，1998

謝辞

本論文を結ぶにあたり，日ごろより懇切なるご指導を賜りました倉重健太郎先生に深く感謝の意を表します．また，ご助言，ご指導をいただいた畑中雅彦先生，渡辺修先生，本田先生，佐賀先生に感謝の意を表します．そして，論文の査読や助言をしていただいた認知ロボティクス研究室の木島康隆さん，中南義典さん，梅津祐介さん，北山直樹さん，澁谷和さん，杉本大志さん，高泉昇太郎さん，沼田利伸さん，三浦丈典さんに感謝いたします．

研究業績

[1] Yoshiki Miyazaki, Kentarou Kurashige, " Use of the knowledge which is independence on reward in Reinforcement Learning ", Proceedings of CIRA 2009, pp. 114-117, 2009

[2] Yoshiki Miyazaki, Kentarou Kurashige, " Use of reward – independent knowledge on reinforcement learning for dynamic environment ", Proceedings of ICACSIS 2010, pp. 303-309, 2010

[3] Yoshiki Miyazaki, Kentarou Kurashige, " Estimate of current state based on experience in POMDP for Reinforcement Learning", Proceedings of the seventeenth International Symposium on Artificial life and Robotics (AROB 17th '12), pp. 1135-1138, 2012