

知識システムの実装基礎

—スライドで理解する人工知能技術—

博士(工学) 新谷 虎松
博士(工学) 大園 忠親 共著
博士(情報学) 白松 俊

コロナ社

まえがき

本書は、多くの読者を獲得し高い評価を得ている前著『Javaによる知能プログラミング入門』[†]に關係して、さらに知識システムを実装する上で必須となる関連技術をピックアップして、実用的な内容にしたものである。前著は、具体的なJavaによる知能プログラミング例を通して、人工知能技術のノウハウや先進的なJavaプログラミングを効果的に習得することを目的にしていた。一方、本書は知識システム（人工知能の応用システム）を研究・実装するという視点から、基本的な人工知能技術の実際的な利用に焦点を当てて、最も基礎となる知識（知能情報処理技術）の理解・習得を目的としている。本書の特長は、他の人工知能入門書のように人工知能全般の内容を広く浅く解説するのではなく、実際的な知識システムを実装する上で共通的によく用いられる基礎技術（基礎の基礎）の解説を、わかりやすくコンパクトにまとめた点にある。本書の内容は、大きく分類して

- (1) 探索技術（1章, 2章, 3章）
- (2) 知識表現と推論技術（4章, 5章, 6章）
- (3) 自然言語処理と情報検索技術（7章）

の3分野に分かれる。

本書の内容は、筆者らの担当講義に対応し、教科書として効果的に利用できるように工夫されている。本書の解説は、実際の講義で使用しているスライドを利用して、わかりやすくまとめられている。読者は、本書のスライドを眺めるだけでも、基礎技術の概略を理解・確認することが可能である。具体的に、「探索技術」の内容は、筆者の大畠の担当講義「知能処理学」（学部2年次後期）に

[†] 2002年2月26日に初版を発行し、第6刷まで増刷された。2010年3月15日付で改訂版を発行し、現在、2011年8月10日付の改訂版第2刷を発売中。

対応している。また、「知識表現と推論技術」の内容は、筆者の新谷の担当講義である「知識システム」(学部3年次後期)に対応している。さらに、「自然言語処理と情報検索技術」は、筆者ら(新谷, 大園, 白松)の担当講義である「自然言語処理」(学部3年次後期)に対応している。すなわち、本書は、これら3つの講義のおもな内容が、効果的に包含されている。

最近、知識システム技術の応用として、Webの高度先端的な活用が広く研究されている。おもな研究のキーワードとしては、Webインテリジェンス(知的Web)、Webエージェント、ウィズダムWeb、セマンティックWeb、Webマイニングなどがある。大目標は、Webを利用したWeb人工知能の実現であり、Webを効果的に活用するものである。Webは世界中のデータを取り込んだ単なる巨大な情報源や情報配信メディアとしてではなく、国境のない社会基盤としてより重要な位置を占めるに至っている。Webが世界を変えていくとともに、Webに関する研究トレンドも大きく変わりつつある。社会へのWebの影響が大きくなる上で、秩序あるWebと自由なWebとの間のトレードオフを調整しつつ、社会全体の合意形成を支援するようなWebを実現する必要がある。このためには、情報科学におけるさらなる研究が必要だけでなく、哲学、倫理学、宗教学、言語学、法学、政治学、社会学、経済学など多種多様な分野の研究者が英知を結集して、Web人工知能を実現する必要がある。Web人工知能に関する、基礎理論、要素技術、そしてシステム開発の、さらなる発展が要望される。

2009年10月14日~11月14日、2010年2月23日~3月15日の2回にわたり、経済産業省が「アイデアボックス」という意見募集Webサイトの試験運用を行った。アイデアボックスは、米国のOpen for Questionsの日本版に相当し、Twitterと連動したシステム設計など、一般人が公的機関に直接働きかける仕組みとして意義深い試みである。実際に寄せられた意見はやはり玉石混合だったが、アイデアを投稿したり、自分と似たアイデアにコメントして議論を深めようとする体験は、一般市民が直接参画できるオープンガバメントとはこのようなものだろうかという未来像を予感させてくれるものであった。

運用終了からかなり経過したいまでも、Twitter 上でハッシュタグ #openmeti を検索すると、関連ツイートを見つけることができる。現段階では意見収集が第一の目的であるが、やがて政策決定のための根拠を共有化し合意形成を支援するシステムに発展させていくべきだろう。ここでは、Web 人工知能の理想的な応用として、問題解決能力を持った集合知による合意形成を支援するシステムの構築が期待される。しかし、そのようなシステムへと発展させていくためには、知識システムの構築技術に関連して、まだ多くの学術的・技術的課題が残されている。例えば、多くのステークホルダーが複雑に関係し合う社会には、円滑な合意形成を妨げるさまざまな障害が存在する。合意形成プロセスをモニタリングして障害要因を特定するのは容易ではなく、未だ人間の高度なヒューリスティクスに頼らざるを得ない課題であるが、これらを工学的アプローチで研究するための要素技術は、すでに揃いつつある。特に、多くのステークホルダー間で自然言語を用いて交わされる^{なま}生の議論過程を工学のまな板に載せ、交渉や合意形成という観点から「議論の意味構造とはなにか」を再定義する上で、ソーシャル Web、多人数インタラクション等の「若い」分野が果たす役割は大きい。近い将来、議論や合意形成プロセスをモニタリングあるいは円滑化するための Web 人工知能の基盤技術が次々に開発され、実用化されることが期待できる。本書によって、知識システムの入門者が、実際のシステムを試作するための基礎技術を磨き、現代社会が強く要望する Web 人工知能にチャレンジしながら、人工知能技術やシステムの実装の面白さを実感していただければ幸いである。

各章は以下のような内容である。1 章では、知識システムの最も重要なポイントである探索技術を学ぶことができる。ここには探索技術の基礎が凝縮されており、探索問題の本質である組合せ爆発に対するアプローチを、効果的に習得することが可能である。ここで紹介している各手法のアイデアを丁寧に読んでいけば、探索アルゴリズムのポイントを理解できる。2 章では、ゲームのための探索技術に関連して、将棋や囲碁などの思考型ゲームなどで利用される要素技術について説明する。この章を読めば、3 目並べの思考アルゴリズムを理解

することができる。プログラミングが可能ならば、実際に思考プログラムを作成するとよい。3章で学ぶ制約充足問題は、複雑な問題を定式的に表現するための強力なツールとなる。ここでは、制約充足問題の定式化方法を学び、それを探索により効果的に解くための手法を知ることができる。4章では、人工知能技術の要素技術となる知識表現を示す。ここでは、セマンティックネットおよびフレームを示す。5章では、推論機構として、ルールベースシステム（プロダクションシステム）を理解することができる。6章では、プランニングの基礎知識を示す。7章では、自然言語処理の入門者が容易に理解できるような構成をとって、自然言語処理技術を解説する。この章を読めば、自然言語処理の概要について理解でき、既存のツールを使った自然言語処理が可能になる。具体的な自然言語処理の1つとして、構文解析アルゴリズムを学ぶ。7.4節「情報検索技術」は、テキスト検索システムを自力で作成するための知識が身につくように構成されている。7.5節「機械学習を用いたテキスト分類」では、機械学習ツール Weka によって機械学習を体験することができる。本書では、参考文献を巻末にまとめて整理し、解説文の中では文献を引用しない形式をとっている。これは、読者が Web ブラウザを使って独自に検索・学習することを意図している。学習者が自主的に、疑問に持ったキーワードを調べるとよい。

本書は、大きく3つの分野に分かれる。すなわち、1章から3章までの探索技術関連、4章から6章までの知識表現と推論技術関連、および自然言語処理と情報検索技術関連の7章である。本書は、読者の目的に応じて読み進めることができるように構成されている。本書を授業で使う例を示す。探索アルゴリズム関連の授業で用いる場合は、1章の探索技術に重点を置くとよい。さらに、応用として7章の情報検索を示すと効果的である。自然言語処理関連の授業で用いる場合は、7章に重点を置くとよい。特に、機械学習を用いたテキスト分類は、実際的な応用システムを実装するための基礎知識となる。人工知能関連の授業で用いる場合は、1章から始め、2章および3章を経て6章まで、順に解説を進めるとよい。この場合、推論技術としてのルールベースシステムは重要な基礎知識となる。独学で読み始める読者は、自分が学習したい分野に重点

を置いて学習を進めるとよい。特に、3つ目の分野の自然言語処理関連は、最新の自然言語処理技術の入門書として活用することができる。

最後になりますが、本書の探索技術（1章、2章および3章）の執筆に関連して、名古屋工業大学大学院情報工学専攻の和田山正教授から、担当講義「知能処理学」の講義スライドのご提供をいただきました。本書の探索技術関連の内容は、ご提供いただいたスライドを利用させていただきました。心から感謝いたします。共著者のみなさんとは、本書の完成に向けて、楽しく、有意義な時間を多く持つことができました。深く感謝します。佐野博之君（現 名古屋工業大学大学院後期課程3年次）には、本書全般にわたっての文書ファイルの整理や草稿の作成をお手伝いしていただいた。また、新谷研究室の多くの大学院生のみなさんには、本文で使用するスライドの作成やバグの発見などに協力していただいた。ここに深く感謝いたします。

2012年8月

新谷虎松

目 次

1. 探 索 技 術

1.1 探索による問題解決とは	1
探 索 木	5
探索アルゴリズムの尺度	12
1.2 基本的な探索アルゴリズム	14
横型探索と縦型探索	18
深さ制限探索と反復深化探索	27
繰り返し回避型探索	30
1.3 評価関数に基づく探索アルゴリズム	33
最小コスト優先探索	41
最良優先探索	45
A (A*) アルゴリズム	50
IDA*アルゴリズム	59
1.4 メタヒューリスティクス	60
山 登 り 法	60
ビーム探索法	62
焼き鈍し法	62
タブー探索	63
遺伝的アルゴリズム	63

2. ゲーム木探索

2.1	ゲームとは	64
2.2	ゲーム木	66
2.3	min-max 戦略に基づく評価値計算	75
	確定的ゲームの場合	75
	不確定的ゲームの場合	81
2.4	α - β カット	84
2.5	例題：3目並べ	93

3. 制約充足問題

3.1	制約充足問題とは	97
3.2	基本的な解法	104
	総探索法	104
	バックトラック法	106
3.3	制約ネットワーク	113
	制約伝搬法	118

4. 知識表現技術

4.1	知識とは	133
4.2	セマンティックネット	134
4.3	フレーム	139

5. ルールベースシステム

5.1	ルールベースシステムとは	147
5.2	パターン照合と変数	150
5.3	演繹システムとリアクションシステム	151
5.4	前向き推論	153
5.5	後ろ向き推論	155
5.6	競合解消戦略 LEX と MEA	159
5.7	推論の高速化	161

6. プランニング

6.1	プランニングとは	165
6.2	STRIPS アプローチ	168
6.3	古典的プランニングの実用例	171

7. 自然言語処理技術

7.1	自然言語処理とは	176
7.2	自然言語処理における解析	177
7.3	CYK 法による構文解析	182
7.4	情報検索技術	187
	適合率と再現率	189
	全文検索技術	191
	ベクトル空間モデル	193
	TF - IDF 法	196

7.5 機械学習を用いたテキスト分類	199
決定木	200
サポートベクトルマシン	201
引用・参考文献	205
索引	212

1

探索技術

1.1 探索による問題解決とは

ある問題をコンピュータにより解くことを考える。効率的なアルゴリズムが知られていない場合には、探索技術の利用が有効である。現実世界には効率的なアルゴリズムが知られていない問題が少なくない。また、あらゆる問題に対して万能な探索技術は存在しないので、問題に応じて適切な探索技術を選択することが必要である。それでは、与えられた問題を探索によって解くにはどうすればよいかを考えよう。スライド 1.1 の上半分は、探索による問題解決における 4 つのステップを表している。

最初のステップでは、与えられた問題を探索に適した形式で表現する必要がある。そのための表現形式としては、系の状態をノードとし、状態間の状態遷移をリンク（辺、エッジ）とするグラフ構造、もしくは、木構造が用いられる。特に断りがなければ、リンクに関して向きの有無を気にしないこととする。向きのあるリンクをアーク（有向辺）と呼ぶ。

2 番目のステップでは、グラフ構造もしくは木構造からゴールを探す問題として、スライド 1.1 の下半分に示すように定式化する。定式化に関しては後述する。3 番目のステップでは、問題に応じたグラフ探索もしくは木探索アルゴリズムを準備する。探索アルゴリズムを実装し、実行するまでの準備を行う。最後のステップでは、実際にプログラムを実行し、解を出力させる。

探索による問題解決

- 探索による問題解決のステップ
 1. 状態をノードとする、グラフ構造・木構造として表現
 2. その構造からゴールを探す問題として定式化
 3. 問題に応じたグラフ探索・木探索アルゴリズムを準備
 4. アルゴリズムを実行

- 問題の定式化
 - 状態 (state) と初期状態 (initial state)
 - 行為 (action)
 - 次の状態 (successor) への遷移を表わす
 - 状態と行為により状態空間 (state space) ができる
 - ゴール検査 (goal test)
 - 経路コスト (path cost)

スライド 1.1 探索による問題解決

探索による問題解決のステップ2において、コンピュータで問題を解くために、与えられた問題を定式化する必要がある。具体的には、つぎの状態、初期状態、行為、ゴール検査、および、経路コスト、の5つを定める必要がある。状態は、与えられた問題におけるある時点での様子を表す。例えば、この問題における状態が1つの変数 x で表現できるならば、 $x = 1$ や $x = 2$ のように具体的な値を割り当てたものは状態であるし、値を割り当てない場合も1つの状態であるといえる。

初期状態 (initial state) は、与えられた問題における最初の状況を表す状態である。行為 (action) は、ある状態からつぎの状態 (successor) への遷移を表す。状態空間 (state space) とは、初期状態から行為による状態遷移により到達可能なすべての状態を表すグラフ構造である。その問題が解決可能な問題であれば、問題を解決した状態 (ゴール、目標状態) も状態空間に含まれる。ゴー

8パズル

初期状態

2	3	5
7	1	6
4	8	

→

目標状態

1	2	3
4	5	6
7	8	

左

2	3	5
7	1	6
4		8

↑

上

2	3	5
7	1	
4	8	6

- 状態：8枚のタイルと1つの空白の場所
- 初期状態：任意の状態の内の1つ
- 行為：空白を上下左右に移動
 - 後者関数：空白を移動した状態を生成
- ゴール検査：目標状態との一致を検査
- 経路コスト：経路のステップ数
各ステップのコストは1

スライド 1.2 8パズル

ル検査 (goal test) は、ある状態が問題を解決した状態、すなわち、ゴールであるかを検査するための手段である。ゴールとなる状態が列挙されている場合もあれば、ある状態がゴールか否かを判定するための関数が与えられる場合もある。経路コスト (path cost) とは、ある状態から別の状態に遷移する際のコストであり、すなわち、与えられた問題におけるある行為の実行に必要なコストである。コストは問題により定義され、金額、時間、距離など多様なコストが考えられる。

問題の定式化について説明するための例題として、スライド 1.2 に示す 8 パズルを考える。8 パズルでは、 3×3 のマス目に 8 枚のタイルと空白が配置される。8 枚のタイルには、それぞれ 1~8 までの数字が書かれている。最初に与えられたこの 8 枚のタイルの配置 (例えばスライドの初期状態) を、特定の配置 (例えばスライドの目標状態) に変換することが、このパズルの目的である。タ

4 1. 探 索 技 術

イルを自由に移動することができれば、この問題は簡単すぎて、パズルとしては面白くない。そこで、問題を難しくするためにタイルの移動方法に関するつぎの2つのルールが与えられている。

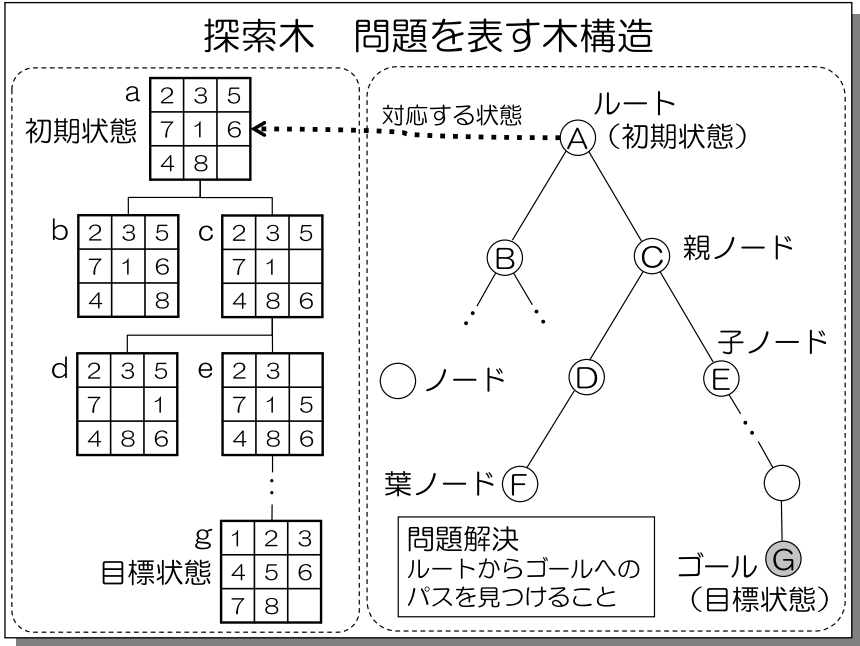
- ルール1：タイルは 3×3 のマスキから外に移動することはできない。
- ルール2：空白へのタイルの移動だけが可能である。

スライド1.2の右上の例を見てみよう。これらの図は初期状態からの盤面の変化を表している。ここでは、タイルを移動させるのではなく、空白を移動させると考えることにする。初期状態において空白を移動可能な方向は、左方向および上方向である。もしタイルの移動として捉えれば、移動可能なタイルは、6および8の2枚のみである。8のタイルは右方向に移動（右上における図の左下の盤面）でき、6のタイルは下方向に移動（右上における図の右下の盤面）できる。

この2つのルールを満たしつつ、初期状態から目標状態にタイルを移動させるには、多くの手順が必要である。人間なら、試行錯誤によりその手順を見つけることができる。ここでは、コンピュータを使ってタイルを移動する手順を見つける方法を考える。

まずは、8パズルを定式化することを考える。先に説明したように、状態、初期状態、行為、ゴール検査、および、経路コストを定めればよい。読者も自分で定式化してみるとよい。

8パズルにおける状態は、 3×3 のマスキに配置された8枚のタイルと1つの空白で表現できる。すなわち、8枚のタイルと空白の 3×3 のマスキ上でのそれぞれの位置である。初期状態は、最初に与えられた8枚のタイルと空白から表現される状態である。任意の状態が初期状態として可能である。行為はパネルの移動である。すなわち、空白の位置を上下左右に移動するようにタイルを移動させ、その結果として行為前の状態からパネルが移動した新たな状態が得られる。ゴール検査では、与えられたゴールと一致するかを検査すればよい。経路コストについては、各行為をコスト1とすれば、パネルを移動した回数がコストとして表現される。ここでは離散的な時間を考えており、時間はステップ



スライド 1.3 探索木

ごとに経過する。1ステップでは1行為が可能であることにする。経路 (path) とは、状態空間内においてある状態から別の状態に遷移する場合に、1つの行為のみで遷移できれば、その行為を表すリンクが経路となる。ある状態 A から別の状態 B に遷移するために、さらに別の状態 V を経由しなければならない場合は、A-V-B が経路となる。ここでの経路のコストとは、経路上のすべてのリンクのコストの和である。すなわち、初期状態から目標状態までのステップ数である。

探索木

初期状態から行為を次々に選択する過程が、探索木 (search tree) として表現される。あるノードにおいて実行可能な複数の行為に対して、その行為を実行した場合の遷移先として、子ノードが展開される。それらの子ノードはさら

に複数の行為を選択肢として持ち、それらの行為を実行した結果として、さらに新たなノードが展開される。展開されたノードがゴールでなく、かつ子を持たないノードであれば、それは手詰まりを表している。この処理を繰り返すことで、いつかはゴールとなるノードが発見される。このようにして、ルートからゴールまでの経路を見つけることが、まさに問題解決である。

8パズルの例では、スライド 1.3 の左側に示すように、初期状態 a において 2 つの行為が実行可能であった。空白を左に移動した状態を b とし、上に移動した状態を c とする。c に対して空白を移動した状態が d および e である。最終的に目標状態 g に到達すれば、8パズルを解く手順がわかったことになる。

スライド 1.3 の右側は、この 8パズルの探索木を表している。探索木は、問題を表す木構造である。探索木におけるノードは元の問題における 1 つの状態を表している。この探索木では、円がノードを表している。例えば、ノード A は初期状態 a を表している。初期状態 a から状態 b および c への遷移は、探索木におけるノード B および C により表現される。

初期状態 a から状態 b への遷移、すなわち、「空白を左に動かす」という行為は、探索木においてノード A からノード B へのリンクとして表現される。図ではノード間の線分がリンクを表している。

ノードは親子関係を持つ。例えば、ノード C は子ノード D および E を持つといい、また、ノード B および C は親ノード A を持つという。親ノードを持たないノードをルート（ルートノード）と呼び、子ノードを持たないノードを葉（葉ノード）と呼ぶ。

知的エージェントに対しても、パズルと同じような問題として考えることができる。スライド 1.4 は知的エージェントの模式図である。知的エージェントは、ゴールを達成するためにアクションの列を適切に選択する必要がある。エージェントは、知覚を使った情報収集により環境から現在の状態を取得して、複数の可能なアクションの中から最も好ましいアクションを選択し、そのアクションを実行することによって、環境の状態およびエージェントの内部状態を変えていく。

索引

【あ】

アーク (arc) 1
アーク無矛盾 (arc consistency) 121, 123
アーク無矛盾性維持 (maintaining arc consistency) 128
アーク無矛盾性チェックアルゴリズム (arc consistency checking algorithm) 124
アサーション (assertion) 147

【い】

遺伝的アルゴリズム (genetic algorithm) 63
意味解析 (semantic analysis) 180
意味表現 (semantic representation) 180
インスタンス (instance) 141
インスタンスフレーム (instance frame) 141
インタプリタ (interpreter) 147

【う】

後ろ向き推論 (backward chaining) 155

【え】

枝刈り (pruning) 201
エッジ (edge) 1
演繹システム (deduction system) 151

【お】

オープンテスト (open test) 199
オープンリスト (open list) 14, 16
オントロジー (ontology) 180

【か】

下位概念 (sub class) 135
係り受け解析 (dependency analysis) 179
係り受け木 (dependency tree) 178
格 (case) 180
確定的ゲーム (deterministic game) 75
隠れ変数 (hidden variable) 114
過剰適合 (over fitting) 199
完全ゲーム木 (complete game tree) 66
完全情報 (perfect information) 65
完全性 (completeness) 13

【き】

機械学習 (machine learning) 199
機能語 (function word) 195
キュー (queue) 22
競合解消戦略 (conflict resolution) 152, 159
競合ルール (conflict rule) 152

教師あり学習 (supervised learning) 199
教師なし学習 (unsupervised learning) 199
近傍探索 (neighborhood search) 60

【く】

句 (phrase) 179
空間計算量 (space complexity) 14
句構造規則 (phrase structure rule) 182, 184
組合せ最適化問題 (combinatorial optimization problem) 103
クラス (class) 141
クラスフレーム (class frame) 141
クローズドテスト (closed test) 199
訓練データ (training data) 199

【け】

形態素 (morpheme) 177
形態素解析 (morphological analysis) 177
系列 (sequence) 165
経路 (path) 5
経路コスト (path cost) 3
ゲーム (game) 64
ゲーム木 (game tree) 66
決定木 (decision tree) 200

検索システム (information retrieval system) 187

【こ】

行為 (action) 2, 165
 後件 (consequent) 147, 149
 膠着語 (agglutinative language) 177
 構文解析 (syntactic analysis) 178
 構文解析器 (parser) 179
 構文木 (parse tree) 178
 ゴール (goal) 2
 ゴール検査 (goal test) 2
 語義曖昧性解消 (word sense disambiguation) 181
 コスト (cost) 35, 40
 古典的プランニング (classical planning) 171

【さ】

再現率 (recall) 189, 190
 再現率-適合率グラフ (recall-precision graph) 190
 最小コスト優先探索 (uniform-cost search) 40, 41
 最適解 (optimal solution) 12
 最適性 (optimality) 13
 最良優先探索 (best-first search) 40, 45
 索引語 (index term) 191, 194
 —の重み (term weight) 196

サブクラス (sub class) 140
 サポートベクトルマシン (support vector machine) 201

【し】

時間計算量 (time complexity) 14

事実 (fact) 147
 辞書規則 (lexical rule) 183, 185
 自然言語処理 (natural language processing) 176
 実体 (entity) 180
 質問応答 (question answering) 187
 修辞関係 (rhetorical relation) 182
 終端記号 (terminal symbol) 184

主辞 (head) 180
 手段目標解析法 (means-ends analysis) 168
 述語項構造解析 (predicate argument structure analysis) 180
 上位概念 (super class) 135
 照応 (anaphora) 182
 照応解消 (anaphora resolution) 182
 状態 (state) 165
 状態空間 (state space) 2
 状態遷移系 (state transition system) 7
 情報伝搬 (information propagation) 135
 情報要求 (information needs) 188
 情報利得比 (information gain ratio) 201
 初期状態 (initial state) 2

【す】

スーパークラス (super class) 140
 スタック (stack) 22
 スロット (slot) 139

【せ】

生成 (generate) 9, 14

生成テスト法 (generate and test) 104
 制約 (constraint) 97
 制約充足問題 (constraint satisfaction problem) 97
 制約ソルバー (constraint solver) 103
 制約伝搬法 (constraint propagation) 118
 制約ネットワーク (constraint network) 113
 接辞処理 (stemming) 194
 セマンティックネット (semantic net) 134
 ゼロ照応 (zero anaphora) 182
 前件 (antecedent) 147, 149
 宣言的知識 (declarative knowledge) 133
 選択 (select) 9, 14
 選択制限 (selectional restriction) 181
 全文検索 (full-text search) 188

【そ】

総探索法 (exhaustive search, brute-force search) 104
 属性継承 (inheritance) 135, 136

【た】

縦型探索 (depth-first search) 18
 タブー探索 (tabu search) 63
 短期記憶 (short-term memory) 152
 探索木 (search tree) 5
 談話関係 (discourse relation) 182

【ち】

知識 (knowledge) 133

長期記憶 (long-term memory) 152
 チョムスキー標準形 (Chomsky normal form) 182

【つ】

つぎの状態 (successor) 2

【て】

定義域 (domain) 97
 適合率 (precision) 189, 190
 テストデータ (test data) 199

手続き的知識 (procedural knowledge) 133
 デモン手続き (demon procedure) 141
 展開 (expand) 9, 14

【と】

統率 (govern) 180
 特徴選択 (feature selection) 201

【な】

内容型検索 (content-based search) 188
 内容語 (content word) 195

【に】

認識-行動サイクル (recognize-act cycle) 148, 153
 認知科学 (cognitive science) 152

【の】

ノード (node) 1

【は】

葉 (leaf) 6
 パターン照合 (pattern matching) 148, 150

バックトラック法 (backtracking) 106
 幅優先探索 (breadth-first search) 18
 反復深化探索 (iterative-deepening search) 29

【ひ】

ビーム探索法 (beam search) 62
 非終端記号 (nonterminal symbol) 184
 ヒューリスティクス (heuristics) 33
 ヒューリスティック関数 (heuristic function) 38, 40

評価関数 (evaluation function) 37, 75
 評価値 (estimated cost) 34

【ふ】

ファクト (fact) 147
 ファシット (facet) 139
 ブーリアンモデル (boolean model) 192
 不確定的ゲーム (probabilistic game) 81
 深さ制限探索 (depth-limited search) 28
 深さ優先探索 (depth-first search) 18
 不完全情報 (imperfect information) 65
 不要語 (stop word) 194
 プランニング (planning) 165
 フレーム (frame) 139
 プロダクション (production) 147

プロダクションシステム (production system) 147, 152

文書単語マトリクス (term-document matrix) 191, 195
 文書ベクトル (document vector) 194
 文生成 (sentence generation) 182
 文節まとめ上げ (chunking) 179

文脈解析 (contextual analysis) 181

【へ】

ベクトル空間モデル (vector space mode) 193
 変数 (variable) 97

【ま】

前終端記号 (preterminal symbol) 184
 前向き推論 (forward chaining) 153
 前向きチェック (forward checking) 119
 マッチング (matching) 150

【も】

目標状態 (goal state) 2
 問題分割法 (problem reduction) 168

【や】

焼き鈍し法 (simulated annealing) 62
 山登り法 (hill climbing) 60

【ゆ】

ユニファイ (unify) 150
 ユニフィケーション (unification) 150

【よ】

横型探索 (breadth-first search) 18

【り】

リアクションシステム (reaction system) 151, 152

リスト (list) 16

理想評価関数 (ideal evaluation function) 38, 39

領域 (domain) 97

リンク (link) 1

【る】

類似度 (similarity) 194

ルート (root) 6

ルール (rule) 147

ルールインタプリタ (rule interpreter) 147

ルールベース (rule base) 147

ルールベースシステム (rule-base system) 147

【わ】

ワーキングメモリ (working memory) 147

【A】

A アルゴリズム (A algorithm) 40, 50

A*アルゴリズム (A* algorithm) 40, 53

AC-1 124

AC-3 131

AND/OR 木 (AND/OR tree) 70

【B】

bag-of-words 191

【C】

CSP (constraint satisfaction problem) 97

CYK 構文解析表 (CYK chart) 185

CYK 法 (Cocke-Younger-Kasami algorithm) 182

C4.5 201

【F】

F 値 (F-measure) 190

FIFO (first-in-first-out) 22

【G】

GA (genetic algorithm) 63

【I】

IDA*アルゴリズム (iterative-

deepening A* algorithm) 60

IDF (inverse document frequency) 198

if 部 (if part) 147, 149, 166

if-add-delete 165

【L】

LEX (lexicographic sort) 159, 160

LHS (left hand side) 149

LIFO (last-in-first-out) 22

【M】

MAC (maintaining arc consistency) 128

MEA (means-ends analysis sort) 159, 160

min-max 戦略 (min-max strategy) 66, 76

【N】

N-クイーン問題 (N-Queens problem) 100

N 分割交差検定 (N-fold cross validation) 200

【R】

Rete アルゴリズム (Rete algorithm) 162

RHS (right hand side) 149

【S】

SHRDLU 187

STRIPS 165

STRIPS アプローチ (STRIPS approach) 165

SVM (support vector machine) 201

【T】

TF (term frequency) 197

TF-IDF 196

then 部 (then part) 147, 149

【W】

Weka 201, 203

~~~~~

2 項制約 (binary constraint) 113

2 値分類器 (binary classifier) 201

~~~~~

α カット (alpha pruning) 87

α - β カット (alpha-beta pruning) 66, 84

β カット (beta pruning) 89

— 著者略歴 —

新谷 虎松（しんたに とらまつ）

- 1980年 東京理科大学理工学部経営工学科卒業
- 1982年 東京理科大学大学院理工学研究科修士課程修了（経営工学専攻）
- 1982年 富士通株式会社 国際情報社会科学研究所
- ～93年 （現、株式会社富士通研究所）勤務
- 1992年 博士（工学）（東京理科大学）
- 1993年 名古屋工業大学助教授
- 1999年 名古屋工業大学教授
- 2003年 名古屋工業大学大学院教授（情報工学専攻）
現在に至る
- 1999年～2000年 米国カーネギーメロン大学客員研究員
- 2004年 大学発ベンチャー 株式会社ウイズダムウェブ設立創業者、代表取締役
CEO（研究成果活用兼業）
- 2008年 社団法人情報処理学会フェロー

大園 忠親（おおぞの ただちか）

- 1995年 名古屋工業大学工学部知能情報システム学科卒業
- 1997年 名古屋工業大学大学院理工学研究科博士前期課程修了（電気情報工学専攻）
- 2000年 名古屋工業大学大学院理工学研究科博士後期課程修了（電気情報工学専攻）
- 2000年 博士（工学）（名古屋工業大学）
- 2000年 名古屋工業大学助手
- 2006年 名古屋工業大学大学院助教授（情報工学専攻）
- 2007年 名古屋工業大学大学院准教授（情報工学専攻）
現在に至る
- 2004年 大学発ベンチャー 株式会社ウイズダムウェブ設立創業者、取締役副社長
CTO（研究成果活用兼業）
- 2004～05年 マレーシ亚马ルチメディア大学工学部客員研究員

白松 俊（しらまつ しゅん）

- 2000年 東京理科大学理工学部情報科学科卒業
- 2003年 東京理科大学大学院理工学研究科修士課程修了（情報科学専攻）
- 2003年 産業技術総合研究所サイバーアシスト研究センター CREST 研究補助員
- 2004年 産業技術総合研究所情報技術研究部門 CREST 研究補助員
- 2008年 京都大学大学院情報学研究科博士課程修了（知能情報学専攻）
- 2008年 博士（情報学）（京都大学）
- 2008年 京都大学大学院情報学研究科学振特別研究員（PD）
- 2009年 名古屋工業大学大学院助教（情報工学専攻）
現在に至る

知識システムの実装基礎

—スライドで理解する人工知能技術—

Foundations of Intelligent Knowledge Systems

—Understanding Artificial Intelligence Techniques Using Lecture Slides—

© Shintani, Ozono, Shiramatsu 2012

2012年10月22日 初版第1刷発行

★

検印省略

著者 新谷虎松
大園忠親
白松俊

発行者 株式会社 コロナ社
代表者 牛来真也

印刷所 三美印刷株式会社

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844 ・ 電話 (03) 3941-3131 (代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-02465-4 (新井) (製本: 愛千製本所) G

Printed in Japan



本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられております。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めておりません。

落丁・乱丁本はお取替えいたします