

CC2001 CS

コンピューティングカリキュラム 2001 コンピュータ科学

付録 A CS 知識体系

(IEEE-CS ACM)

翻訳第 1 版

2007 年 7 月 31 日

情報処理学会 CS 教育委員会

石畑 清 (明治大学, 幹事)

大岩 元 (慶應義塾大学)

清水 謙多郎 (東京大学)

長崎 等 (共栄大学)

中谷 多哉子 (筑波大学)

疋田 輝雄 (明治大学, 委員長)

箕原 辰夫 (千葉商科大学)

渡辺 治 (東京工業大学)

板野 肯三 (筑波大学)

角田 博保 (電気通信大学)

玉井 哲雄 (東京大学)

中里 秀則 (早稲田大学)

野中 誠 (東洋大学)

三浦 孝夫 (法政大学)

和田 耕一 (筑波大学)

付録A

CS知識体系

コンピューティングカリキュラム 2001 レポートの付録 A は、コンピュータ科学の大学学部生のカリキュラムにおいて教えられることを想定した知識領域を定義する。この分類構想のもととなる基本的な考え方、およびその歴史的経緯、構造、応用に関する詳細は、タスクフォースレポート本文に含まれる。この付録はレポート全文よりも広く流布すると思われるので、タスクフォースは、推薦カリキュラムを理解するために必要な基本的概念の要約を各付録に入れることが重要であると考え、最も重要な概念を以下に要約する。

知識体系の構造

CS の知識体系は階層的に三つのレベルで構成される。最も高いレベルの階層は領域 (area) であり、特定の学問分野の部分を表す。各領域は、オペレーティングシステムを表す OS、プログラミング言語を表す PL のように、2 文字の略記号によって識別される。領域は、個々の主題のモジュールを表すユニットと呼ばれる小さい区分に分解される。各ユニットは、領域名の後に数字を付加することによって識別される。例えば OS3 は並行性に関するユニットである。各ユニットはさらに最も低いレベルの階層であるトピックスの集まりに細分化される。

コアと選択のユニット

CC1991 において確立した枠組みから知識体系を改訂するにあたり、CC2001 タスクフォースは、コンピューティングの分野が拡大し、大学生がかつてはその分野に不可欠であると考えられたすべての科目を学ぶことが不可能になったという事実を考慮しなければならなかった。そこでタスクフォースは、コンピュータ科学を教えるすべての人が、この分野で学部卒業資格を得るどの学生にとっても本質的に不可欠なものであると認める教科から成る最小限のコアを定義しようと試みた。大学学部生のプログラムの一部として提供されるが、コアから外れる教科は選択科目とした。コアの定義において広いコンセンサスが必要であると主張するタスクフォースは、コアをできる限り小さくすることを望んでいる。これによって、教育機関は、その個々のニーズを満たすやり方でカリキュラムの選択科目を自由に組み立てることができる。

CC2001 推薦カリキュラムの開発の過程におけるわれわれの議論において、次の点を強調することが有益であることがわかった。

* コアは完結したカリキュラムではない。コアは最小限のものと定義されるので、それだけでは完結した学部のカリキュラムを構成しない。すべての学部生のプログラムは知識体系から選択ユニットを追加しなければならない。ただし CC2001 レポートはどのユニットを選ぶべきかまでは決めていない。

* コアユニットは、学部生のカリキュラムの初めに設定される入門コースとは限らない。コアとされたユニットの多くは、実際入門的な科目であるが、その分野について予め重要な予備知識を得た後でなければ履修できないコアユニットも存在する。例えば、タスクフォースは、すべての学生がある程度意味のあるアプリケーションシステムを開発しなければならないと考える。この規模のプ

プロジェクトを首尾よく管理するために不可欠な教科は、すべての学生にその履修が要求されているので、したがってコアの一部となる。一方、プロジェクトを行うコースは、学部学生のプログラムの終わりごろに経験をすることになる可能性が高い。また、初級コースはコア教科をカバーするとともに選択ユニットを取り入れることも可能である。コアの指定は単に必修科目を意味するだけであり、それを扱うコースのレベルとは関係がない。

ユニットを履修するのに必要とされる時間の評価

特定のユニットを履修するのに必要とされる時間の見当を読者に与えるために、CC2001 レポートは測定の基準を定める尺度を定義しなければならない。しかし世界中で認められる標準的な尺度が存在しなかったため、このような測定基準を選ぶことは困難であることが判明した。これまでのカリキュラムレポートとの一貫性を保つために、タスクフォースは時間数で時間を表現することを決定した。これは伝統的な講義形式で授業を行うのに必要とされる授業時間に対応したものである。しかし、予想されるいかなる混乱をも一掃するために、目安として教室における時間を使用することについての次の観点を強調しておくことは重要である。

* タスクフォースは講義形式を推奨しようとするものではない。伝統的な講義形式に由来する測定基準を使っているが、タスクフォースは、同様に効果的な他のスタイル、教育技術において最近特に改善をみているようなものもあると考える。これらのスタイルのいくつかに対しては、時間を単位とするという考えが適用しにくいかもしれない。としても、時間についてのこの指示は、授業スタイルとは関係なしに、少なくとも相対的な尺度の手段として役立つであろう。例えば、5時間かかるユニットをカバーするためには、1時間かかるユニットの約5倍の時間をおそらく必要とするであろうという意味である。

* 指定された時間は教室の外で費やされる時間を含まない。ユニットに割り当てられた時間は、教員の準備時間や学生が授業時間外で費やす時間を含んでいない。一般的なガイドラインとして、教室の外での学習時間は授業時間の約3倍である。したがって、3時間と指定されているユニットは、通常、約12時間（教室内3時間、教室外9時間）を必要とする。

* ユニットに記載された時間は、カバーするための最小限の時間を表す。各ユニットに割り当てた時間数は、学生がそのユニットの学習目標を達成できるための必要最小時間であると解釈すべきである。そのユニットに指示された最小時間以上を費やすことは常に適切である。

CSの知識体系の詳細

CS 知識体系の要約を図 A-1 に示す。この中には、領域、ユニット、いずれのユニットがコアか、コアユニットに対する最小要求時間が示してある。各領域の詳細は以下の個々の節に記載される。

図 A-1 コンピュータ科学知識体系 (*印はコアユニットを示す)

DS 離散構造 (43 コア時間)

- * DS1 関数, 関係, 集合 (6)
- * DS2 論理の基礎 (10)
- * DS3 証明技法 (12)
- * DS4 数え上げの基礎 (5)
- * DS5 グラフと木 (4)
- * DS6 離散確率 (6)

PF プログラミングの基礎 (38 コア時間)

- * PF1 プログラミングの基本的構成要素 (9)
- * PF2 アルゴリズムと問題解決 (6)
- * PF3 基本データ構造 (14)
- * PF4 再帰 (5)
- * PF5 イベント駆動プログラミング (4)

AL アルゴリズムと計算の複雑さ (31 コア時間)

- * AL1 アルゴリズムの解析の基礎 (4)
- * AL2 アルゴリズム設計手法 (6)
- * AL3 基礎的な計算アルゴリズム (12)
- * AL4 分散アルゴリズム (3)
- * AL5 計算可能性の基礎 (6)
- AL6 計算量クラス P と NP
- AL7 オートマトン理論
- AL8 アルゴリズムの高度な解析
- AL9 暗号アルゴリズム
- AL10 幾何アルゴリズム
- AL11 並列アルゴリズム

AR アーキテクチャと構成 (36 コア時間)

- * AR1 論理回路と論理システム (6)
- * AR2 データのマシンレベルでの表現 (3)
- * AR3 アセンブリレベルのマシン構成 (9)
- * AR4 メモリシステムの構成とアーキテクチャ (5)
- * AR5 インタフェースと通信 (3)
- * AR6 機能的構成 (7)
- * AR7 並列処理と新たなアーキテクチャ (3)
- AR8 性能の向上
- AR9 ネットワークと分散システムのためのアーキテクチャ

OS オペレーティングシステム (18 コア時間)

- * OS1 オペレーティングシステムの概要 (2)
- * OS2 オペレーティングシステムの原理 (2)
- * OS3 並行性 (6)
- * OS4 スケジューリングとディスパッチ (3)
- * OS5 メモリ管理 (5)
- OS6 デバイス管理
- OS7 セキュリティと保護
- OS8 ファイルシステム
- OS9 リアルタイムシステムと組込みシステム
- OS10 フォールトトレランス
- OS11 システム性能評価
- OS12 スクリプティング

NC ネットワークコンピューティング (15 コア時間)

- * NC1 ネットワークコンピューティング入門 (2)
- * NC2 通信とネットワーク接続 (7)
- * NC3 ネットワークセキュリティ (3)
- * NC4 クライアントサーバコンピューティングの例としてのウェブ (3)
- NC5 ウェブアプリケーションの構築
- NC6 ネットワーク管理
- NC7 データ圧縮
- NC8 マルチメディアデータ技術
- NC9 ワイヤレスおよびモバイルコンピューティング

PL プログラミング言語 (21 コア時間)

- * PL1 プログラミング言語の概要 (2)
- * PL2 仮想計算機 (1)
- * PL3 言語翻訳入門 (2)
- * PL4 宣言と型 (3)
- * PL5 抽象化メカニズム (3)
- * PL6 オブジェクト指向プログラミング (10)
- PL7 関数型プログラミング
- PL8 言語翻訳システム
- PL9 型システム
- PL10 プログラミング言語の意味論
- PL11 プログラミング言語の設計

HC ヒューマンコンピュータインタラクション (8 コア時間)

- * HC1 ヒューマンコンピュータインタラクションの基礎 (6)
- * HC2 簡単なグラフィカル・ユーザインタフェースの構築 (2)
- HC3 人間中心のソフトウェア評価
- HC4 人間中心のソフトウェア開発
- HC5 グラフィカル・ユーザインタフェースの設計
- HC6 グラフィカル・ユーザインタフェースのプログラミング
- HC7 マルチメディアシステムの HCI 的側面
- HC8 協同作業とコミュニケーションの HCI 的側面

GV グラフィックスとビジュアル・コンピューティング (3 コア時間)

- * GV1 グラフィックスにおける基礎技術 (2)
- * GV2 グラフィック・システム (1)
- GV3 グラフィック・コミュニケーション
- GV4 幾何モデリング
- GV5 レンダリングの基本
- GV6 高度なレンダリング
- GV7 高度な技術
- GV8 コンピュータ・アニメーション
- GV9 視覚化
- GV10 仮想現実
- GV11 コンピュータ・ビジョン

IS インテリジェントシステム (10 コア時間)

- * IS1 インテリジェントシステムの基本的問題 (1)
- * IS2 探索および制約充足 (5)
- * IS3 知識表現および推論 (4)
- IS4 高度な探索
- IS5 高度な知識表現と推論
- IS6 エージェント
- IS7 自然言語処理
- IS8 機械学習とニューラルネット
- IS9 AI プランニングシステム
- IS10 ロボット工学

IM 情報管理 (10 コア時間)

- * IM1 情報モデルとシステム (3)
- * IM2 データベースシステム (3)
- * IM3 データモデリング (4)
- IM4 関係データベース
- IM5 データベース問合わせ言語
- IM6 関係データベース設計
- IM7 トランザクション処理
- IM8 分散データベース
- IM9 データベースの物理設計
- IM10 データマイニング
- IM11 情報格納と検索
- IM12 ハイパーテキストとハイパーメディア
- IM13 マルチメディア情報とシステム
- IM14 電子図書館

SP 社会的な観点と職業専門人としての問題 (16 コア時間)

- * SP1 コンピュータの歴史 (1)
- * SP2 コンピュータの社会的文脈 (3)
- * SP3 分析の方法およびツール (2)
- * SP4 専門上のまた倫理的な責任 (3)
- * SP5 コンピュータ・ベース・システムのリスクと脆弱性 (2)
- * SP6 知的財産権 (3)
- * SP7 プライバシーと市民的自由 (2)
- SP8 コンピュータ犯罪
- SP9 コンピュータにおける経済問題
- SP10 哲学的枠組み

SE ソフトウェア工学 (31 コア時間)

- * SE1 ソフトウェア設計 (8)
- * SE2 API の使用 (5)
- * SE3 ソフトウェアツールおよび環境 (3)
- * SE4 ソフトウェアプロセス (2)
- * SE5 ソフトウェア要求および仕様 (4)
- * SE6 ソフトウェア妥当性検査 (3)
- * SE7 ソフトウェアの進化 (3)
- * SE8 ソフトウェアプロジェクト管理 (3)
- SE9 コンポーネントベース開発
- SE10 形式手法
- SE11 ソフトウェアの信頼性
- SE12 専用システムの開発

CN 計算科学と数値計算 (コア時間なし)

- CN1 数値解析
- CN2 オペレーションズリサーチ
- CN3 モデリングとシミュレーション
- CN4 ハイパフォーマンス・コンピューティング

注：括弧内の数字は、内容を講義形式でカバーするのに要する最低限の時間を表す。
これ以上の時間を費やすことは常に適切である。

離散構造 (DS)

- DS1 . 関数, 関係, 集合 [コア]
- DS2 . 論理の基礎 [コア]
- DS3 . 証明技法 [コア]
- DS4 . 数え上げの基礎 [コア]
- DS5 . グラフと木 [コア]
- DS6 . 離散確率 [コア]

離散構造はコンピュータ科学の基礎となる題材である。ここで「基礎」と呼んでいるのは、離散構造を主に研究するコンピュータ科学者は比較的少数であること、けれども、離散構造に関する概念を利用する能力が、コンピュータ科学の多くの分野で必要であること、などの理由による。離散構造には、集合論、論理、グラフ理論、そして組合せ論などの重要な概念が含まれる。

離散構造の題材は、データ構造やアルゴリズムなどの分野ではいたるところに見られるが、コンピュータ科学の他の分野でも登場してくる。例えば、形式的な証明を作成したり理解する能力は、形式仕様、プログラムの検証、暗号理論などにおいて本質的である。また、グラフの概念は、ネットワーク、オペレーティングシステム、コンパイラなどで使われる。集合の概念は、ソフトウェア工学、データベースなどで使われる。

コンピュータ科学が分野として成熟してくるに従い、実際的な問題を処理するために、これまでより一層洗練された解析技術が導入されてくるようになる。こうした将来のコンピュータ科学の技術を理解するために、現在の学生たちには、離散構造についての確かな基礎知識が必要となるだろう。

最後に、一般に分野の境は曖昧になりがちだが、このことは特に離散構造において顕著である、ということを指摘しておこう。われわれは、ここに数学的な題材で、コンピュータ科学の教育で教えるべきもの、そしてコンピュータ科学の教育者たちが詳細まで教えることができるほど熟知しておくべきものを集めてきた。けれども、この領域 (DS) やアルゴリズムと計算の複雑さ (AL) の領域と基礎となる数学として残したトピックスとの間に引く境界線についての判断は、どうしても一意に決めることはできない。読者には、これらの二つの領域の必須項目の中で、学校によっては、離散構造などといった名称の授業で教えられているものもあることを注意しておこう。

DS1 関数, 関係, 集合 [コア]

最低履修時間:6 時間

トピックス:

関数

- 全射
- 単射
- 逆関数
- 合成

関係

- 反射律
- 対称律
- 推移律
- 同値関係

集合

- ベン図
- 補集合
- デカルト積
- べき集合

鳩の巣原理

濃度と可算性

学習の目標:

1. 関数, 関係, 集合などに関する基本用語を例を用いて説明する。
2. 集合, 関数, 関係に関する操作を行う。
3. 実例的な例を, 集合, 関数, 関係などの適切なモデルに関連付ける。さらに関連する操作や概念をその中で正しく当てはめる。
4. 基本的な数え上げ論法を示す。その中には, 対角線論法や鳩の巣原理などを含む。

DS2 論理の基礎 [コア]

最低履修時間:10 時間

トピックス:

命題論理

論理記号

真理値表

標準形 (和積標準形と積和標準形)

恒真性

述語論理

全称記法, 存在記法

三段論法と対偶

述語論理の限界

学習の目標:

1. 命題記号論理ならびに述語論理の形式的手法を利用する。
2. 記号論理における形式的手法がアルゴリズムや現実の状況をモデル化するのにどのように使われるかを説明する。
3. 形式的論理証明や論理的な推論を用いてパズルなどの問題を解く。

4. 述語論理の重要性, そしてその限界を説明する。

DS3 証明技法 [コア]

最低履修時間:12 時間

トピックス:

含意, 逆, 否定, 対偶, 否定, 矛盾, などの概念
形式的証明の構造
直接的な証明
反例による証明
対偶による証明
背理法
数学的帰納法
強帰納法
再帰的定義
整列順序

学習の目標:

1. このユニットで示される証明技法の各々に対し, その基本構造の概要を述べ, 証明の例を挙げる。
2. 与えられた問題に対し, どの証明技法が最適であるかを議論する。
3. 数学的帰納法を再帰や再帰的に定義された構造に適用する。
4. 数学的帰納法と強帰納法の違いを述べ, それぞれに対して適切な利用例を挙げる。

DS4 数え上げの基礎 [コア]

最低履修時間:5 時間

トピックス:

数え上げ論法

- 和と積の法則
- 包除原理
- 算術数列と幾何数列
- フィボナッチ数

鳩の巣原理
順列と組合せ

- 基本定義
- パスカルの三角形
- 二項定理

漸化式の解法

- 典型例
- 分類定理

学習の目標:

1. 順列や組合せを計算し, 具体的な応用の中で, それらの意味を理解する。
2. 分類定理の定義を述べる。
3. 様々な基本漸化式を解く。
4. 問題を解析し, それを表現する漸化式を作成する。あるいは, 問題の中にある数え上げ問題を認識する。

DS5 グラフと木 [コア]

最低履修時間:4 時間

トピックス:

木
無向グラフ
有向グラフ
全域木
巡回戦略

学習の目標:

1. グラフ理論の基本用語や, その各々に対する性質のいくつかや特殊例について, 例を用いて説明する。
2. 木やグラフ上の巡回方法を複数示す。
3. コンピュータ科学上の問題を, グラフや木を用いてモデル化する。
4. グラフや木をデータ構造, アルゴリズム, あるいは数え上げに関係付ける。

DS6 離散確率 [コア]

最低履修時間:6 時間

トピックス:

有限の確率空間, 確率測度, 事象
条件付き確率, 独立性, ベイズの定理
整数値の確率変数, 期待値

学習の目標:

1. 簡単なゲームなどの初歩的な問題に対し, 事象の確率や確率変数の期待値などを計算す

- る。
2. 従属事象と独立事象を区別する。
 3. 二項定理を独立事象に適用する。従属事象に対してはベイズの定理を適用する。
 4. 確率的手法を, モンテカルロ法, あるいは, アルゴリズムやハッシュの平均的な解析などの問題に適用する。
-

プログラミングの基礎 (PF)

- PF1 . プログラミングの基本的構成要素 [コア]
- PF2 . アルゴリズムと問題解決 [コア]
- PF3 . 基本データ構造 [コア]
- PF4 . 再帰 [コア]
- PF5 . イベント駆動プログラミング [コア]

プログラミング言語に習熟することは、コンピュータ科学の学習にとって必要不可欠である。CC1991 報告書において、プログラミング言語の知識は必須であると認識されていたが、カリキュラム中ではほとんど強調されなかった。CC1991 報告書の中の「プログラミング言語入門」領域では、「ますます多くの学生が……中学校においてそのような経験を得ている」との楽観的な仮定のもとに、教室での授業への割当てをわずか 12 時間だけにし、しかもこれを選択科目としていた。われわれは、大学生のコンピュータ科学プログラムにおいて、少なくとも一つのプログラミング言語の使い方を十分に学生に教えなければならないと確信する。われわれは、さらに、コンピュータ科学プログラムが少なくとも二つのプログラミングパラダイムを利用した言語に堪能になるように、学生に教えるべきであると推奨する。この目的の達成のためには 12 時間をはるかに超える時間が必要である。

この知識分野は、基礎となるパラダイムとは独立に、プログラミングの実践に不可欠な技術および概念から成り立っている。したがって、この領域 (PF) は、基本的プログラミング概念、基本データ構造、およびアルゴリズム的プロセスに関するユニットを含んでいる。しかし、これらのユニットだけでは、コンピュータ科学の学生が習得しなければならないすべての範囲のプログラミング知識をカバーすることはできない。多くの他の領域 (特にプログラミング言語 (PL) およびソフトウェア工学 (SE)) もまた、学部学生のコアの一部であるプログラミング関連のユニットを含んでいる。これらのユニットは、大部分のケースにおいて「プログラミングの基礎」か、あるいはさらに上級の領域のどちらかに割り当てられてもよかったであろう。

PF1 プログラミングの基本的構成要素 [コア]

最低履修時間:9 時間

トピックス:

- 高水準言語の基本構文と意味論
- 変数, 型, 式, 代入
- 単純な入出力
- 条件判定と繰返しの制御構造
- 関数と引数受渡し
- 構造的分解

学習の目標:

1. このユニットで述べられているプログラミングの基本的構成要素を含んだ簡単なプログラム

の動作を分析し、説明する。

2. 標準的な条件判定と繰返しの制御構造と関数を使用した、短いプログラムを修正し、拡張する。
3. 基本演算、単純な入出力、標準的な条件判定と繰返しの構造、関数の定義といった、プログラミングの基本的構成要素を使用したプログラムを設計し、実現、テスト、デバッグする。
4. 与えられたプログラミングタスクのために、適切な条件判定と繰返しの制御構造を選択する。
5. プログラムをより小さな部分に分割するために、構造的（機能的）分解技術を適用する。
6. 引数受渡しの仕組みを説明する。

PF2 アルゴリズムと問題解決 [コア]

最低履修時間:6 時間

トピックス:

問題解決戦略
問題解決過程におけるアルゴリズムの役割
アルゴリズムの実現戦略
デバッグ戦略
アルゴリズムの概念と特性

学習の目標:

1. 問題解決過程におけるアルゴリズムの重要性について議論する。
2. よいアルゴリズムに必要とされる特性を識別する。
3. 簡単な問題を解くためのアルゴリズムを作り出す。
4. 簡単な問題を解くために、擬似コードまたはプログラミング言語を使ってアルゴリズムを実現、テスト、デバッグする。
5. デバッグに役立つ戦略を説明する。

PF3 基本データ構造 [コア]

最低履修時間:14 時間

トピックス:

基本型
配列
レコード
文字列と文字列処理
メモリ内でのデータの表現
静的割当て、スタック割当て、ヒープ割当て
実行時記憶管理

ポインタと参照

連結構造

スタック, キューおよびハッシュ表の実現戦略

グラフと木の実現戦略

適切なデータ構造を選択するための戦略

学習の目標:

1. 基本データ型と組み込みデータ構造の表現および使用について議論する。
2. トピックスに示した各データ構造がメモリ内でどのように割り当てられ、使われるかを説明する。
3. トピックスに示した各データ構造に対する典型的な応用を説明する。
4. 高水準言語においてユーザ定義のデータ構造を実現する。
5. データ構造をいくつか実現し、効率の観点からそれらを比較する。
6. 次の各データ構造を使うプログラムを書く: 配列, レコード, 文字列, 連結リスト, スタック, キュー, ハッシュ表。
7. データ構造の動的実現と静的実現のコストと効果について比較対照する。
8. 与えられた問題をモデル化するための適切なデータ構造を選択する。

PF4 再帰 [コア]

最低履修時間:5 時間

トピックス:

再帰の概念

再帰的数学関数

簡単な再帰的手続き

分割統治法

再帰的バックトラック法

再帰の実現

学習の目標:

1. 再帰の概念を説明し、その使用例を示す。
2. 再帰的に定義された問題の基底ケースと一般化ケースを識別する。
3. 階乗のような初歩的な問題の反復解法と再帰的解法を比較する。
4. 分割統治法を説明する。
5. 簡単な再帰的関数や手続きを実現、テスト、デバッグする。
6. スタックを用いた再帰の実現法を説明する。
7. バックトラック法が適切な解法となる問題について議論する。
8. ある問題に対してどんな場合に再帰的解法が適しているかを判断する。

PF5 イベント駆動プログラミング [コア]

最低履修時間:4 時間

トピックス:

イベント処理手法

イベント伝播

例外処理

学習の目標:

1. イベント駆動プログラミングとコマンドラインプログラミングの違いを説明する。
 2. ユーザのイベントに応える簡単なイベント駆動プログラムを設計し、コーディング、テスト、デバッグする。
 3. 実行時に起きる例外条件に反応するコードを開発する。
-

アルゴリズムと計算の複雑さ (AL)

- AL1 . アルゴリズムの解析の基礎 [コア]
- AL2 . アルゴリズム設計手法 [コア]
- AL3 . 基礎的な計算アルゴリズム [コア]
- AL4 . 分散アルゴリズム [コア]
- AL5 . 計算可能性の基礎 [コア]
- AL6 . 計算量クラス P と NP [選択]
- AL7 . オートマトン理論 [選択]
- AL8 . アルゴリズムの高度な解析 [選択]
- AL9 . 暗号アルゴリズム [選択]
- AL10 . 幾何アルゴリズム [選択]
- AL11 . 並列アルゴリズム [選択]

アルゴリズムはコンピュータ科学とソフトウェア工学の基礎である。実世界のどんなソフトウェアシステムでも、性能はただ二つの要因によって決まる。すなわち、(1) どんなアルゴリズムを用いたか、(2) その実現の様々な段階において妥当で効率的なものを選んだか、の 2 点である。したがって、よいアルゴリズムの設計は、すべてのソフトウェアシステムの効率にとって本質的である。アルゴリズムの学習は、問題の可能な解決技法だけでなく、問題の本質についての洞察法も与えてくれる。このアルゴリズム的な問題解決の技法は、プログラミング言語、プログラミングパラダイム、コンピュータハードウェア、等々、アルゴリズムの実現の様々な側面とは独立なものである。

計算を設計するのに重要な要素は、具体的な目的に対して適切なアルゴリズムを選び、それを適用する能力である。それには適当なアルゴリズムが存在しない場合もあるという認識も必要である。こうした能力は、基本的な主要問題群を扱うアルゴリズムの種類を知ること、それらの長所と短所、そして特定の状況における有効性を認識すること、などに依存する。効率はこの分野を通して常に考えられている課題である。

AL1 アルゴリズムの解析の基礎 [コア]

最低履修時間:4 時間

トピックス:

計算量の上界, 平均計算量に対する漸近的解析
最良, 平均, 最悪時の振る舞いの違いの区別
大きな O 記法, 小さな o 記法, オメガ記法, シータ記法
標準的な計算量クラス
性能の実験的な測定
アルゴリズムにおける時間と領域のトレードオフ
漸化式を用いた再帰的アルゴリズムの解析

学習の目標:

1. 大きな O 記法, オメガ記法, そしてシータ記法を用いて, アルゴリズムによる仕事量を記述する方法を説明する。
2. 大きな O 記法, オメガ記法, そしてシータ記法を用いて, アルゴリズムの時間計算量と領域計算量に対する漸近的な上界, 下界, あるいは上下界を与える。
3. 単純なアルゴリズムの時間計算量と領域計算量を求める。
4. 再帰的に定義されたアルゴリズムの時間計算量に対する再帰的な関係式を導く。
5. 初歩的な漸化式を解く。

AL2 アルゴリズム設計手法 [コア]

最低履修時間:6 時間

トピックス:

カズクアルゴリズム
 貪欲アルゴリズム
 分割統治法
 バックトラック法
 分枝限定法
 ヒューリスティックス
 パターン照合アルゴリズム, 文字列・テキスト処理アルゴリズム
 近似数値アルゴリズム

学習の目標:

1. カズクアルゴリズムの欠点を指摘する。
2. 各々のアルゴリズム (カズク法, 貪欲法, 分割統治法, バックトラック法, 分枝限定法, ヒューリスティックス) に対し, 日常の人間の行動でその基本概念を示す例を挙げる。
3. 代表的な問題に対し, 貪欲アルゴリズムでプログラムを作成する。
4. 代表的な問題に対し, 分割統治法でプログラムを作成する。
5. バックトラック法を用いて迷路探索のような問題を解く。
6. 問題解決法としての様々なヒューリスティックスを記述する。
7. 文字列照合を用いて部分文字列を解析する。
8. 近似数値計算を用いて多項式の根を求めるなどの数学的問題を解く。

AL3 基礎的な計算アルゴリズム [コア]

最低履修時間:12 時間

トピックス:

単純な数値計算アルゴリズム
 逐次探索と二分探索アルゴリズム
 $O(N^2)$ 時間整列アルゴリズム (選択整列, 挿入整列)

O(N log N) 時間整列アルゴリズム (クイックソート, ヒープソート, マージソート)
ハッシュ表と衝突回避法
二分探索木
グラフの表現 (隣接リスト, 隣接行列)
深さ優先探索, 幅優先探索
最短経路アルゴリズム (ダイクストラのアルゴリズム, フロイドのアルゴリズム)
推移閉包 (フロイドのアルゴリズム)
最小全域木 (プリムのアルゴリズム, クラスカルのアルゴリズム)
トポロジカルソート

学習の目標:

1. 最も一般的な $O(N^2)$ 時間と $O(N \log N)$ 時間の整列アルゴリズムのプログラムを作成する。
2. 応用のため適切なハッシュ関数を設計し実現する。
3. ハッシュ表の衝突解決アルゴリズムを設計し実現する。
4. 整列, 探索, ハッシュに対する主要なアルゴリズムの計算効率について議論する。
5. 計算効率以外でアルゴリズムの選択に影響を与える要因, 例えば, プログラミングに要する時間, 保守性, 応用に依存した入力データの特性, などについて議論する。
6. 種々の問題をグラフに関する基本的なアルゴリズムを用いて解決する。例えば, 深さ優先探索, 幅優先探索, 単一出発点最短経路, 全対最短経路, 推移閉包, トポロジカルソート, そして, 少なくとも 1 種類の最小全域木アルゴリズム, などである。
7. アルゴリズムを評価する。可能な選択肢の中から適切なアルゴリズムを選択し, その妥当性を説明する。アルゴリズムをプログラムとして実現する。

AL4 分散アルゴリズム [コア]

最低履修時間: 3 時間

トピックス:

合意形成とリーダー選出
終了判定
フォールトトレランス
安定性

学習の目標:

1. 分散計算のパラダイムを説明する。
2. 単純な分散アルゴリズムを少なくとも一つ説明する。
3. 合意形成アルゴリズムやリーダー選出アルゴリズムを用いるべき場合を決める。
4. 論理的時計と物理的時計を区別する。
5. 分散アルゴリズムにおける各事象の相対的順序を説明する。

AL5 計算可能性の基礎 [コア]

最低履修時間:6 時間

トピックス:

有限状態機械
文脈自由文法
実質的に計算可能・不可能な問題
計算不可能関数
停止問題
計算不可能性から導かれること

学習の目標:

1. 有限状態機械の概念を議論する。
2. 文脈自由文法について説明する。
3. 与えられた言語を受理する決定性有限状態機械を設計する。
4. ある種の問題に対してアルゴリズム的な解決が不可能な理由を説明する。
5. 計算不可能性の概念を説明する例を挙げる。

AL6 計算量クラス P と NP [選択]

トピックス:

クラス P と NP の定義
NP 完全性 (クックの定理)
標準的な NP 完全問題
還元技法

学習の目標:

1. クラス P と NP の定義を述べる。
2. NP 完全性の重要性を説明する。
3. 問題の NP 完全性をよく知られた NP 完全問題からの還元によって証明する。

AL7 オートマトン理論 [選択]

トピックス:

決定性有限オートマトン (DFA)
非決定性有限オートマトン (NFA)
DFA と NFA の等価性
正規表現

正規表現に対するポンプの補題
プッシュダウンオートマトン (PDA)
PDA と文脈自由文法 (CFG) の関係
文脈自由文法の性質
チューリング機械
非決定性チューリング機械
集合と言語
チョムスキー階層
チャーチ・チューリングの提唱

学習の目標:

1. 与えられた言語のチョムスキー階層での位置付け (正規集合, 文脈自由, 文脈依存, 帰納的枚挙可能言語) を判断する。
2. ある言語がある特定のクラスに所属し, 一つ下のクラスには所属しないことを証明する。
3. 言語クラスごとに, 同値だが異なる表現方法の間の変換をする。例えば, DFA, NFA, 正規表現の 3 者間, あるいは, PDA と CFG など。
4. トップダウンとボトムアップの構文解析のそれぞれに対し, 少なくとも一つのアルゴリズムを説明する。
5. チャーチ・チューリングの提唱とその重要性を説明する。

AL8 アルゴリズムの高度な解析 [選択]

トピックス:

ならし解析 (amortized analysis)
オンラインとオフライン・アルゴリズム
乱択アルゴリズム
動的計画法
組合せ最適化

学習の目標:

1. ポテンシャル法を用い, これまでに現れなかったデータ構造に対し, 与えられたポテンシャル関数をもとに, ならし解析を行う。
2. オンライン・アルゴリズムの解析における競合解析の妥当性を説明する。
3. 決定性アルゴリズムが知られていなかったり, あるいは, 既存の決定性アルゴリズムでは計算量が大きくなる問題に対するアルゴリズムの設計において, ランダム性の活用法を説明する。
4. 動的計画法によってアルゴリズムを設計し, プログラムを作成する。

AL9 暗号アルゴリズム [選択]

トピックス:

暗号の歴史の概観
秘密鍵暗号と鍵交換問題
公開鍵暗号
デジタル署名
情報セキュリティプロトコル
応用 (零知識証明, 認証など)

学習の目標:

1. 効率的で基本的な整数論的アルゴリズムを記述する。例えば, 最大公約数, 法 n のもとでの乗法の逆元, 法 n のもとでのべき乗, などの計算である。
2. 少なくとも一つの公開鍵暗号を, その安全性に必要な計算量的仮定を含めて記述する。
3. 既存のプロトコルや暗号の基本要素を用い, 暗号プロトコルの簡単な拡張を作る。

AL10 幾何アルゴリズム [選択]

トピックス:

線分: 性質, 交点
凸包計算アルゴリズム

学習の目標:

1. 少なくとも二つの凸包計算アルゴリズムを記述し, その計算時間を解析する。
2. 凸包計算の計算時間の下界が $\Omega(N \log N)$ であることを示す。
3. そのほかに少なくとも一つの計算幾何学的アルゴリズムを記述する。例えば, 最近頂点对, 凸層, 極大層, などの計算である。

AL11 並列アルゴリズム [選択]

トピックス:

PRAM モデル
メモリ読み書きにおける排他性と同時性
ポインタ・ジャンプ
ブレントの定理と作業効率

学習の目標:

1. リスト構造の PRAM 上の実現を示す。
2. 並列接頭句操作を用いて単純な計算を並列に効率的に行う。
3. ブレントの定理とその関連を説明する。

アーキテクチャと構成 (AR)

- AR1 . 論理回路と論理システム [コア]
- AR2 . データのマシンレベルでの表現 [コア]
- AR3 . アセンブリレベルのマシン構成 [コア]
- AR4 . メモリシステムの構成とアーキテクチャ [コア]
- AR5 . インタフェースと通信 [コア]
- AR6 . 機能的構成 [コア]
- AR7 . 並列処理と新たなアーキテクチャ [コア]
- AR8 . 性能の向上 [選択]
- AR9 . ネットワークと分散システムのためのアーキテクチャ [選択]

情報処理の中心にはコンピュータが存在する。コンピュータがなければ、情報処理のほとんどの分野は、数学的理論の一部に過ぎないであろう。今日、情報処理が関係する分野で専門家になるろうとする人は、コンピュータをプログラムを実行する不思議なブラックボックスだと思ってすませることはできない。すべての学生は、計算機システムの機能的要素や、その性質、性能、相互作用などについて、一定の理解と認識を得る必要がある。これには、実用的な意味もある。学生は、現実のマシンの上でより効率的に動くプログラムを作ることができるようになるために、計算機アーキテクチャを理解する必要がある。使うべきシステムを選ぶ状況においても、CPU のクロックスピードとメモリ容量の間といった種々の構成要素間のトレードオフを理解していることが必要である。

これらの内容に関連して学ぶべきことは、専門的に学習しようとするカリキュラム向けのコアに対応しており、また計算機アーキテクチャを最小の 36 時間のみで履修するカリキュラムにも使用できるように考えられている。最小以上に履修することを前提としたカリキュラムでは、この同じ内容 (AR1 ~ AR7) を二つのコースに分けてより専門的なレベルで取り扱うことができる。さらに選択的なトピックスを取り入れたいときは、この二つのコースの中に取り入れるか、3 番目のコースを用意して、もっと包括的にいろいろなトピックスが取り扱えるようにしてもよい。

AR1 論理回路と論理システム [コア]

最低履修時間:6 時間

トピックス:

計算機アーキテクチャの概要と歴史

基本的構成要素

- 論理ゲート
- フリップフロップ
- カウンタ
- レジスタ
- PLA

論理式, 最小化, 積和形

レジスタ転送表現

物理的特性

- ゲートの遅延
- ファンイン
- ファンアウト

学習の目標:

1. 真空管から VLSI までの計算機アーキテクチャの進歩を説明する。
2. 基本的構成要素と、それらが計算機アーキテクチャの歴史的発展の中でどのような役割を果たしてきたかを理解し、説明する。
3. 簡単な組合せ論理回路や順序回路の機能を数学的表現で表す。
4. 基本的構成要素を用いて簡単な回路を設計する。

AR2 データのマシンレベルでの表現 [コア]

最低履修時間:3 時間

トピックス:

ビット, バイト, ワード

数値表現と基数 (base)

固定小数点と浮動小数点

符号付き絶対値表現と 2 の補数表現

非数値データの表現

- 文字コード
- グラフィックデータ

レコードと配列の表現

学習の目標:

1. 数値データを表現するのに異なる形式が必要な理由を説明する。
2. 負数を、符号付き絶対値表現や 2 の補数表現で表す方法を説明する。
3. 数値データをある形式から別の形式に変換する。
4. 数値の固定長表現が正確さや精度に与える影響について論じる。
5. 非数値データの内部表現について説明する。
6. 文字, 文字列, レコード, 配列の内部表現について説明する。

AR3 アセンブリレベルのマシン構成 [コア]

最低履修時間:9 時間

トピックス:

フォンノイマン計算機の基本構成

制御ユニット

- 命令フェッチ
- デコード
- 実行

命令セットと命令の種類

アセンブリ／機械語プログラミング

命令の形式

アドレッシングモード

サブルーチンのコールとリターンのメカニズム

入出力と割込み

学習の目標:

1. 古典的なフォンノイマン計算機の構成と主要な機能ユニットについて説明する。
2. 古典的なフォンノイマン計算機で命令がどのように実行されるかを説明する。
3. 命令が、2進のマシンレベルとアセンブラの記号レベルで、どのように表現されるか要点を説明する。
4. 命令ごとのアドレス、可変長命令と固定長命令など、様々な命令形式について説明する。
5. アセンブリ言語によって簡単なプログラムを書く。
6. 高水準言語の基本要素が機械語のレベルでどのように実現されるか説明する。
7. サブルーチンコールがアセンブリ言語のレベルでどのように取り扱われるかを説明する。
8. 割込みや入出力の基本概念を説明する。

AR4 メモリシステムの構成とアーキテクチャ [コア]

最低履修時間:5 時間

トピックス:

記憶システムとこれを実現するための技術

コーディング, データ圧縮, データの完全性

メモリ階層

主記憶の構成と操作

レイテンシ, サイクルタイム, バンド幅, インターリーブ

キャッシュメモリ

- アドレス割当て
- ブロックサイズ
- 置換えと書込みに関するポリシー

仮想記憶

- ページテーブル
- TLB

フォールト処理と信頼性

学習の目標:

1. 主要なメモリ技術について説明する。
2. メモリのレイテンシがプログラムの実行時間に及ぼす影響について説明する。
3. メモリの実効レイテンシを減少させるための記憶階層の利用について説明する。
4. メモリ管理の原理について説明する。
5. キャッシュや仮想記憶の役割について説明する。
6. 仮想記憶の機構をもつシステムの動作について説明する。

AR5 インタフェースと通信 [コア]

最低履修時間:3 時間

トピックス:

入出力の基礎

- ハンドシェイク
- バッファリング
- プログラム入出力
- 割込みを用いた入出力

割込みの構造

- 割込みベクトル
- 優先順位
- 割込みに対する応答

外部記憶, 物理的構成, ドライブ

バス

- バスプロトコル
- 排他制御
- ダイレクトメモリアクセス (DMA)

ネットワークの入門的説明

マルチメディアのサポート

RAID アーキテクチャ

学習の目標:

1. 入出力の制御やデータの転送に割込みがどのように使われるか説明する。
2. コンピュータの中で使われている様々なバスについて説明する。
3. 磁気ディスク装置からのデータアクセスについて説明する。
4. よく使われるネットワークの構成について比較し, 論じる。
5. マルチメディアのサポートに必要なインタフェースについて説明する。
6. RAID アーキテクチャの利点と限界について説明する。

AR6 機能的構成 [コア]

最低履修時間:7 時間

トピックス:

簡単なデータパスの構成
制御ユニット

- ワイヤードロジックによる実現
- マイクロプログラムによる実現

命令のパイプライン化

命令レベル並列性 (ILP) の概説

学習の目標:

1. データパスの複数の構成について比較し論じる。
2. 各構成要素に必要な制御, ワイヤードロジックやマイクロプログラムによる制御信号の生成について論じる。
3. パイプラインによる基本的な命令レベル並列性と, 起こりうるハザードについて説明する。

AR7 並列処理と新たなアーキテクチャ [コア]

最低履修時間:3 時間

トピックス:

SIMD, MIMD, VLIW, EPIC の紹介

シストリックアーキテクチャ

内部結合ネットワーク

- ハイパーキューブ
- シャッフル
- メッシュ
- クロスバ

共有メモリスistem

キャッシュのコヒーレンス

メモリモデルと一貫性

学習の目標:

1. 古典的フォンノイマンモデルを超えた並列処理の概念について論じる。
2. SIMD, MIMD, VLIW などの新しいアーキテクチャについて説明する。
3. 内部結合ネットワークの概念と種々の方法の特徴について説明する。
4. メモリ管理に関して並列処理システムが内包する問題点と対策について論じる。

AR8 性能の向上 [選択]

トピックス:

スーパースカラーアーキテクチャ
分岐予測
プリフェッチ
投機的実行
マルチスレッディング
スケーラビリティ

学習の目標:

1. スーパースカラーアーキテクチャとその長所について説明する。
2. 分岐予測とその有効性について説明する。
3. プリフェッチのコストと有効性について説明する。
4. 投機的実行について説明し、正しい実行を保証するための条件について論じる。
5. マルチスレッディングが提供する性能的優位性について、アーキテクチャの観点から論じる。また、理想的な性能向上を困難にする要因について論じる。
6. スケーラビリティと性能の関係を説明する。

AR9 ネットワークと分散システムのためのアーキテクチャ [選択]**トピックス:**

LAN と WAN の紹介
階層的プロトコル設計, ISO/OSI, IEEE 802
分散アルゴリズムにおけるアーキテクチャ的要素の影響
ネットワークコンピューティング
分散マルチメディア

学習の目標:

1. ネットワークの基本要素について説明し、LAN と WAN の違いを明確に説明する。
 2. 階層的ネットワークプロトコルの設計に関係するアーキテクチャ的要素について論じる。
 3. ネットワークシステムや分散システムでは、アーキテクチャがどのように変わってくるかを説明する。
 4. ネットワークコンピューティングや分散マルチメディアに関連したアーキテクチャ上の諸問題について論じる。
-

オペレーティングシステム (OS)

- OS1 . オペレーティングシステムの概要 [コア]
- OS2 . オペレーティングシステムの原理 [コア]
- OS3 . 並行性 [コア]
- OS4 . スケジューリングとディスパッチ [コア]
- OS5 . メモリ管理 [コア]
- OS6 . デバイス管理 [選択]
- OS7 . セキュリティと保護 [選択]
- OS8 . ファイルシステム [選択]
- OS9 . リアルタイムシステムと組込みシステム [選択]
- OS10 . フォールトトレランス [選択]
- OS11 . システム性能評価 [選択]
- OS12 . スクリプティング [選択]

オペレーティングシステムは、ハードウェアの動作の抽象を定義し、プログラマがハードウェアを制御できるようにする。オペレーティングシステムは、また、コンピュータユーザ間の資源の共有を管理する。この領域 (OS) のトピックスは、最新のオペレーティングシステムの設計に影響を与えている諸問題を説明するものである。この領域のコースは、学生にオペレーティングシステムを使った実験を行わせる実習を含むのが一般的になるであろう。

ここ何年かにわたって、オペレーティングシステムとその抽象概念は、典型的なアプリケーションソフトウェアにとって複雑になってきた。内部の実現アルゴリズムやデータ構造の詳細を学ぶ前に、オペレーティングシステムの広範な利用について学生が理解できるようにすることが必要である。したがって、これらの科目では、オペレーティングシステムの利用 (外部) およびその設計と実現 (内部) の両方を扱う。オペレーティングシステム利用にかかわるアイデアの多くは、並行プログラミングなど、コンピュータ科学の分野全般にわたって適用できるものである。内部設計を学ぶことは、ディペンダブルプログラミング、アルゴリズムの設計と実現、最新のデバイス開発、仮想環境の構築、ウェブでのコンテンツのキャッシング、セキュアで安全なシステムの構築、ネットワーク管理など広範な分野に関係する。

OS1 オペレーティングシステムの概要 [コア]

最低履修時間:2 時間

トピックス:

- オペレーティングシステムの役割と目的
- オペレーティングシステム開発の歴史
- 典型的なオペレーティングシステムの機能
- クライアントサーバモデルやハンドヘルドデバイスを支援する機構
- 設計上の問題
 - 効率性

- 頑強性
- 柔軟性
- 可搬性
- セキュリティ
- 互換性

セキュリティ, ネットワーキング, マルチメディア, ウィンドウの影響

学習の目標:

1. 最新のオペレーティングシステムの目的と機能を説明する。
2. 初歩的なバッチシステムから複雑なマルチユーザシステムに至るまで、オペレーティングシステムがどのように進化してきたかを述べる。
3. オペレーティングシステムの設計に伴うトレードオフを分析する。
4. 使いやすさ, 効率性, 進化能力に関係した、最新のオペレーティングシステムの機能を述べる。
5. ネットワーク, クライアントサーバおよび分散オペレーティングシステムについて説明し、それらがシングルユーザオペレーティングシステムとどのように違うかについて論じる。
6. オペレーティングシステムに対する潜在的脅威とそれらを防御するためのセキュリティ機能を見つけ出す。
7. オープンソースソフトウェアやインターネットの利用の増加がオペレーティングシステムの設計にどのような影響を与えているか述べる。

OS2 オペレーティングシステムの原理 [コア]

最低履修時間:2 時間

トピックス:

構成法 (モノリシック, 階層型, モジュール化, マイクロカーネルモデル)
抽象, プロセス, 資源
アプリケーションプログラムインタフェース (API) の概念
アプリケーションからの要求, ハードウェア・ソフトウェア技術の発展
デバイスの構成
割込み: 方式と実現
ユーザ状態・システム状態の概念と保護, カーネルモードへの移行

学習の目標:

1. 論理層の概念を説明する。
2. 階層的に抽象層を構築することの利点を説明する。
3. API とミドルウェアの必要性を主張する。
4. 計算資源が、アプリケーションソフトウェアによっていかに利用され、システムソフトウェアによっていかに管理されているか述べる。

5. オペレーティングシステムのカーネルモードとユーザモードを比較対照する。
6. 割り込み処理を用いることの利点と欠点を論じる。
7. オブジェクト指向, モジュール化, マイクロカーネル, 階層化などのオペレーティングシステムの様々な構成法を比較対照する。
8. デバイスリストとドライバ入出力キューの利用について説明する。

OS3 並行性 [コア]

最低履修時間:6 時間

トピックス:

状態と状態図

データ構造 (実行可能リスト, プロセス制御ブロックなど)

ディスパッチングとコンテキストスイッチング

割り込みの役割

並行実行: 利点と欠点

「相互排除」問題とその解

デッドロック: 要因, 条件, 回避

モデルと機構 (セマフォ, モニタ, 条件変数, ランデブー)

生産者消費者問題と同期

マルチプロセッシングの問題 (スピンロック, 再入可能性)

学習の目標:

1. オペレーティングシステムの枠組みの中で並行性の必要性を説明する。
2. 多くの独立したタスクの並行動作によって生じる実行時の潜在的な問題を示す。
3. 並行処理システムを実現するためにオペレーティングシステムレベルで使用可能な一連の機構をまとめ, それぞれの利点を述べる。
4. タスクがとりうる状態と多数のタスクの管理に必要なデータ構造について説明する。
5. オペレーティングシステムにおいて相互排除問題を解決するための様々なアプローチを要約する。
6. オペレーティングシステムで並行性を支援するのに, 割り込み, ディスパッチング, コンテキストスイッチングが用いられる理由を述べる。
7. 簡単な問題ドメインに対して, 状態および状態遷移図を作成する。
8. 並行性の管理における, スタック, キューなどのデータ構造の有用性について論じる。
9. デッドロック状態が発生する条件について説明する。

OS4 スケジューリングとディスパッチ [コア]

最低履修時間:3 時間

トピックス:

プリエンプティブおよびノンプリエンプティブスケジューリング
スケジューラとスケジューリングポリシー
プロセスとスレッド
デッドラインとリアルタイム問題

学習の目標:

1. 優先度, 性能比較, フェアシェア方式など, オペレーティングシステムのタスクのプリエンプティブ, ノンプリエンプティブ双方のスケジューリングで共通に用いられているアルゴリズムを比較対照する。
2. スケジューリングアルゴリズムとアプリケーションドメインの関係について述べる。
3. 短期, 中期, 長期, 入出力などのプロセッサスケジューリングの分類について論じる。
4. プロセスとスレッドの違いを述べる。
5. リアルタイムスケジューリングに対する静的および動的アプローチについて比較対照する。
6. プリエンプションとデッドラインスケジューリングの必要性について論じる。
7. スケジューリングアルゴリズムに取り入れられた論理が, ディスク入出力, ネットワークスケジューリング, プロジェクトスケジューリング, その他計算処理以外の問題にどのように適用できるか見つけ出す。

OS5 メモリ管理 [コア]

最低履修時間:5 時間

トピックス:

物理メモリとメモリ管理ハードウェアの復習
オーバレイ, スワッピング, パーティション
ページングとセグメンテーション
配置と置換えのポリシー
ワーキングセットとスラッシング
キャッシング

学習の目標:

1. メモリ階層とそのコスト・性能のトレードオフについて説明する。
2. 仮想記憶の概念とそれがハードウェアおよびソフトウェアによってどのように実現されているか説明する。
3. 仮想記憶の原理を, キャッシング, ページング, セグメンテーションへの適用を通して要約する。
4. メモリサイズ (主記憶, キャッシュメモリ, 補助メモリ) とプロセッサ速度におけるトレードオフを評価する。
5. タスクに対するメモリ割当ての各種方式について, それぞれの相対的な長所に触れ, 妥当性を主張する。

6. キャッシュメモリの存在意義とその利用について述べる。
7. ページングとセグメンテーションの技術を比較対照する。
8. スラッシングの概念について、それが発生しうる理由と、その問題を認識し制御するのに用いられる技術について論じる。
9. オーバレイ、スワッピング、配置および置換えポリシーなど、様々なメモリ領域分割技術を解析する。

OS6 デバイス管理 [選択]

トピックス:

シリアルデバイスとパラレルデバイスの特性
デバイスの差異の抽象化
バッファリング方式
ダイレクトメモリアクセス
障害からの回復

学習の目標:

1. シリアルデバイスとパラレルデバイスの主な違いを説明し、それぞれが適する条件を見つけ出す。
2. 物理ハードウェアと、オペレーティングシステムが提供する仮想デバイスの関係を見つけ出す。
3. バッファリングについて説明し、その実現方式を述べる。
4. 様々なデバイス（ハンドヘルドデバイス、ネットワーク、マルチメディアなど）に対して、コンピュータとの接続に用いられている機構の差異を明らかにし、これらがオペレーティングシステムの設計に及ぼす影響を説明する。
5. ダイレクトメモリアクセスの利点と欠点を述べ、その利用が必要とされる環境について論じる。
6. 障害回復に対する要求を見つけ出す。
7. 考えられるデバイスに対して単純なデバイスドライバを実現する。

OS7 セキュリティと保護 [選択]

トピックス:

システムセキュリティの概要
ポリシーと機構の分離
セキュリティの方式と装置
保護, アクセス, 認証
保護のモデル
メモリ保護
暗号化

回復管理

学習の目標:

1. 保護とセキュリティの必要性, コンピュータ利用における倫理の役割について主張する。
2. 保護とセキュリティを提供するのに用いられるオペレーティングシステムの機能と限界について要約する。
3. 現在のセキュリティ実現技術を比較対照する。
4. 現在普及している複数のオペレーティングシステムのセキュリティに関する長所と短所を互いに比較対照する。
5. 現在普及している複数のオペレーティングシステムの回復管理に関する長所と短所を互いに比較対照する。

OS8 ファイルシステム [選択]

トピックス:

ファイル : データ, メタデータ, 操作, 編成, バッファリング, シーケンシャルアクセス, 非シーケンシャルアクセス

ディレクトリ : 内容と構造

ファイルシステム

- パーティショニング
- マウント, アンマウント
- 仮想ファイルシステム

標準的な実現技術

メモリマップトファイル

専用ファイルシステム

ネーミング, 探索, アクセス, バックアップ

学習の目標:

1. ファイルシステムに関する広範な問題を要約する。
2. ファイル編成の各種アプローチを, それぞれの長所と短所を認識しつつ比較対照する。
3. ハードウェアの発展によって, ファイルシステムの設計と管理に関する優先事項がどのように変化してきたかを要約する。

OS9 リアルタイムシステムと組込みシステム [選択]

トピックス:

プロセスおよびタスクのスケジューリング

リアルタイム環境におけるメモリおよびディスク管理に対する要求

障害, リスク, 回復

リアルタイムシステムに特有の問題

学習の目標:

1. リアルタイムシステムとは何かを述べる。
2. リアルタイムシステムにおける遅延の存在を説明し、その特性を述べる。
3. リアルタイムシステムに存在する特有の問題と、これらの問題に対しどのような対応がなされているか要約する。

OS10 フォールトトレランス [選択]

トピックス:

基本概念：高信頼性および高可用性システム
空間的および時間的冗長性
フォールトトレランスの実現方式
高信頼性システムの実例

学習の目標:

1. フォールトトレランス, 信頼性, 可用性の三つの用語の関係を説明する。
2. オペレーティングシステムにおけるフォールトトレランスの様々な実現方式について概要を示す。
3. オペレーティングシステムが、障害発生後もいかに継続して機能することができるか説明する。

OS11 システム性能評価 [選択]

トピックス:

なぜシステム性能評価が必要か
何を評価すべきか
キャッシング, ページング, スケジューリング, メモリ管理, セキュリティなどのポリシー
評価モデル: 決定的, 解析的, シミュレーション, 実装特化
評価データの収集方法 (プロファイルおよびトレース機構)

学習の目標:

1. システムの性能を計測するのに用いられる性能指標について述べる。
2. システムを評価するのに用いられる主要な評価モデルを説明する。

OS12 スクリプティング [選択]

トピックス:

スクリプト言語の記述と役割

基本システムコマンド
スクリプトの作成, 引数受渡し
スクリプトの実行
スクリプティングのプログラミングに与える影響

学習の目標:

1. オペレーティングシステムが提供するシステムコマンドの典型的なセットについて要約する。
 2. スクリプト言語の典型的な機能を挙げ, それらのプログラミングに対する影響を説明する。
 3. スクリプトを実現するための機構, システムの実現と構築におけるスクリプトの役割を挙げる。
 4. 引数受渡しを含んだ簡単なスクリプトを実現する。
-

ネットワークコンピューティング (NC)

- NC1 . ネットワークコンピューティング入門 [コア]
- NC2 . 通信とネットワーク接続 [コア]
- NC3 . ネットワークセキュリティ [コア]
- NC4 . クライアントサーバコンピューティングの例としてのウェブ [コア]
- NC5 . ウェブアプリケーションの構築 [選択]
- NC6 . ネットワーク管理 [選択]
- NC7 . データ圧縮 [選択]
- NC8 . マルチメディアデータ技術 [選択]
- NC9 . ワイヤレスおよびモバイルコンピューティング [選択]

最近のコンピュータおよび通信ネットワーク技術の発達, 特に TCP/IP を用いた技術の発達により, コンピュータ分野におけるネットワーク技術の重要性が増大した。ネットワークコンピューティングは, 以下の項目を含む専門分野をカバーしている: コンピュータ通信ネットワークの概念とプロトコル, マルチメディアシステム, ウェブ標準と技術, ネットワークセキュリティ, ワイヤレスおよびモバイルコンピューティング, 分散処理システム。

この領域 (NC) に精通するためには, 理論と実践の両方の習得が必要である。学生が概念を理解し, またそれを現実問題へ適用できるようにするために, 実験と分析を含む演習を強く推奨する。演習には, データの収集と分析, 実験結果からのモデルの抽出, ソースコードレベルでのプロトコル分析, ネットワークパケットの観察, ソフトウェアの構築, および複数の設計モデルの比較評価を含む。これらはみな, 演習によって最もよく理解されうる重要な概念である。

NC1 ネットワークコンピューティング入門 [コア]

最低履修時間: 2 時間

トピックス:

- ネットワーク化とインターネットの背景と歴史
- ネットワークアーキテクチャ
- ネットワークコンピューティング分野の個別テーマ
 - ネットワークとプロトコル
 - ネットワーク化されたマルチメディアシステム
 - 分散コンピューティング
 - モバイルおよびワイヤレスコンピューティング

学習の目標:

1. 初期のネットワークとインターネットの発展について論じる。
2. 電子メール, telnet, FTP, ニュースグループ, ウェブブラウザ, オンラインウェブ教材, インスタントメッセージといった一般的なネットワークアプリケーションを効果的に使用する能力を示す。

3. 典型的なネットワークアーキテクチャの階層構造を説明する。
4. ネットワークコンピューティング分野における最新技術について説明し、それらの現在の能力、限界、および近い将来の可能性を評価する。

NC2 通信とネットワーク接続 [コア]

最低履修時間:7 時間

トピックス:

ネットワーク標準および標準化団体

ISO 7 層参照モデルの一般論および TCP/IP におけるその具体例

回線交換とパケット交換

ストリームとデータグラム

物理層ネットワーク接続の概念

- 理論的な基礎
- 伝送メディア
- 標準

データリンク層の概念

- フレーミング
- エラー制御
- フロー制御
- プロトコル

ネットワーク間接続とルーティング

- ルーティングアルゴリズム
- ネットワーク間接続
- 輻輳制御

トランスポート層サービス

- 接続確立
- 性能の問題

学習の目標:

1. 重要なネットワーク標準について、歴史的背景の中で論じる。
2. ISO 参照モデルの最初の 4 層の役割を示す。
3. 回線交換とパケット交換について、それぞれの長所と短所を示し、二つの方式の違いについて論じる。
4. ネットワークが伝送エラーを検出し訂正する方法を説明する。
5. パケットがインターネット上でいかにルーティングされるかを例証する。
6. DHCP のような標準のホスト構成設定ソフトウェアツールを使用して、二つのクライアントおよび一つのサーバから成る簡単なネットワークを構成する。

NC3 ネットワークセキュリティ [コア]

最低履修時間:3 時間

トピックス:

暗号の基礎
秘密鍵アルゴリズム
公開鍵アルゴリズム
認証プロトコル
デジタル署名
例

学習の目標:

1. 公開鍵暗号の基本的な考え方について論じる。
2. 公開鍵暗号がいかに機能するかについて述べる。
3. 秘密鍵アルゴリズムと公開鍵アルゴリズムを使い分ける。
4. 一般的な認証プロトコルについてまとめる。
5. PGP キーペアを生成して配布し, PGP パッケージを使用して暗号化した電子メールを送る。
6. 一般に利用可能な暗号手法の能力と限界についてまとめる。

NC4 クライアントサーバコンピューティングの例としてのウェブ [コア]

最低履修時間:3 時間

トピックス:

ウェブ技術

- サーバサイドプログラム
- コモン・ゲートウェイ・インタフェース (CGI) プログラム
- クライアントサイドスクリプト
- アプレットの概念

ウェブサーバの特性

- パーミッションの扱い
- ファイル管理
- 共通サーバアーキテクチャの能力

クライアントコンピュータの役割
クライアントサーバ関係の特徴
ウェブプロトコル
ウェブサイト作成およびウェブ管理のためのサポートツール
インターネット情報サーバの開発
情報やアプリケーションの公開

学習の目標:

1. 各種アプリケーションプログラムのそれぞれについて、クライアントとサーバの役割および責任を説明する。
2. 様々なクライアントサーバ連携の方法を効率的に実現するためのツール群を選択する。
3. 簡単な対話型ウェブベースアプリケーション（例えば、クライアントから情報を集め、それをサーバ上のファイルに格納する簡単なウェブフォーム）を設計し、構築する。

NC5 ウェブアプリケーションの構築 [選択]

トピックス:

アプリケーション層のプロトコル
ウェブ技術の原理
データベース駆動のウェブサイト
遠隔手続き呼出し (RPC)
軽量分散オブジェクト
ミドルウェアの役割
支援ツール
分散オブジェクトシステムにおけるセキュリティの問題
企業内ウェブベースアプリケーション

学習の目標:

1. 対話型クライアントサーバ方式の中規模ウェブアプリケーションを、様々なウェブ技術を用いてどのように作ることができるかを例証する。
2. データベース駆動ウェブサイトをどのように実装するかを示す。この際、アーキテクチャの各層における関連技術、および付随する性能のトレードオフを説明する。
3. 任意の二つの分散オブジェクトフレームワークを使う分散処理システムを実装し、性能とセキュリティの点からそれらを比較する。
4. 企業内ウェブベースアプリケーションにおけるセキュリティの問題およびその対策について議論する。

NC6 ネットワーク管理 [選択]

トピックス:

ネットワーク管理上の課題の概要
パスワードおよびアクセス制御機構の使用
ドメインネームとネームサービス
インターネットサービスプロバイダ (ISP) にかかわる問題
セキュリティとファイアウォール
サービス品質の問題：性能, 障害回復

学習の目標:

1. ウイルス, ワーム, トロイの木馬, サービス不能化攻撃 (DoS) といったセキュリティ上の脅威に起因するネットワーク管理上の問題を説明する。
2. セキュリティ確保のための様々なアプローチについて, その長所と短所をまとめる。
3. 特定の目的のために設計されたシステムにおいて, 適切なレベルのセキュリティを保証する方策を作成する。
4. ネットワークのファイアウォールを実装する。

NC7 データ圧縮 [選択]

トピックス:

アナログ表現およびデジタル表現

符号化と復号のアルゴリズム

可逆的圧縮と不可逆的圧縮

データ圧縮: ハフマン符号とジブ・レンペルのアルゴリズム

オーディオ情報の圧縮と伸張

画像情報の圧縮と伸張

ビデオ情報の圧縮と伸張

性能の問題: タイミング, 圧縮係数, リアルタイム使用への適合性

学習の目標:

1. デジタル表現におけるサンプリングおよび量子化の基礎的な特性をまとめる。
2. テキスト, オーディオ, 画像, およびビデオ情報のために最適な圧縮技術を選択する。その際に, 特定のアプリケーションや特定の状況において考慮すべき要因についても述べる。
3. 圧縮および伸張アルゴリズムの非対称な特性について説明する。
4. 連長符号化の概念を例証する。
5. ハフマン符号やジブ・レンペルのアルゴリズムを使用する UNIX の compress ユーティリティのようなプログラムが, 典型的なテキストファイルをいかに圧縮するかを例証する。

NC8 マルチメディアデータ技術 [選択]

トピックス:

音とオーディオ, 画像とグラフィックス, アニメーションとビデオ

マルチメディア標準

- オーディオ
- 音楽
- グラフィックス
- 画像
- 電話
- ビデオ
- テレビ

容量計画と性能に関する問題

入出力装置

- スキャナ
- デジタルカメラ
- タッチスクリーン
- 音声によって起動する機器

MIDI キーボード, シンセサイザー

記憶装置標準

- 光磁気ディスク
- CD-ROM
- DVD

マルチメディアサーバおよびファイルシステム

マルチメディア開発をサポートするツール

学習の目標:

1. いくつかのメディアまたはマルチメディア標準のそれぞれについて, その標準が要求するものを非技術用語で述べ, その標準のもつ制約を, どの程度人間の知覚が感受することができるかを説明する。
2. あるコンピュータシステムが一つのマルチメディアアプリケーションを提供する能力を, ネットワークに対するマルチメディアシステムの要件も考慮に入れて評価する。
3. マルチメディアアプリケーションの一つを提供するコンピュータシステムのもつべき特性を, 支援ツールと適切な標準も挙げて述べる。
4. 適度の規模のマルチメディアアプリケーションを実装する。

NC9 ワイヤレスおよびモバイルコンピューティング [選択]

トピックス:

ワイヤレス標準の歴史, 発展, 互換性についての概観

ワイヤレスおよびモバイルコンピューティングに固有な問題

無線ローカルエリアネットワークと衛星ネットワーク

ワイヤレスローカルループ

モバイルインターネットプロトコル

移動を意識することによる影響

モバイルに適応するためのクライアントサーバモデルの拡張

モバイルデータアクセス: サーバデータの配布およびクライアントキャッシュの管理

モバイルワイヤレスコンピューティングを支援するソフトウェアパッケージ

ミドルウェアおよび支援ツールの役割

性能の問題

最新技術

学習の目標:

1. モバイル IP の主要な特性を示し, モビリティ管理, 位置管理および性能の点で IP との違いを説明する。
 2. 電子メールや他の通信がモバイル IP を用いてどのように転送されるかを, ホームエージェントおよび外部エージェントを含めて例証する。
 3. モバイルおよびワイヤレスデータ通信に依存する簡単なアプリケーションを実装する。
 4. ワイヤレスおよびモバイルコンピューティングに関して, 現在注目されている, あるいは注目されつつある分野について述べ, それぞれについて, 現在の能力, 限界, および近い将来の可能性を評価する。
-

プログラミング言語 (PL)

- PL1 . プログラミング言語の概要 [コア]
- PL2 . 仮想計算機 [コア]
- PL3 . 言語翻訳入門 [コア]
- PL4 . 宣言と型 [コア]
- PL5 . 抽象化メカニズム [コア]
- PL6 . オブジェクト指向プログラミング [コア]
- PL7 . 関数型プログラミング [選択]
- PL8 . 言語翻訳システム [選択]
- PL9 . 型システム [選択]
- PL10 . プログラミング言語の意味論 [選択]
- PL11 . プログラミング言語の設計 [選択]

プログラミング言語は、プログラマとコンピュータとの主要なインタフェースである。プログラマは単に一つの言語によるプログラミングの方法を知っているだけでなく、様々な言語が提供する異なったスタイルのプログラミングを理解する必要がある。プログラマはその仕事の過程で、同時に多くの異なった言語とスタイルを使うこともあるだろうし、また、一生の間に多くの異なる言語に遭遇するであろう。プログラミング言語の多様性や異なるプログラミングパラダイムの間の設計トレードオフを理解することによって、新しい言語の迅速な習得が容易になる。プログラミング言語の実用的な側面を理解するためには、さらにプログラミング言語の翻訳および実行時機能（記憶域割当てなど）の基礎知識も必要とされる。

PL1 プログラミング言語の概要 [コア]

最低履修時間:2 時間

トピックス:

- プログラミング言語の歴史
- プログラミングパラダイムの簡単なまとめ
 - 手続き型言語
 - オブジェクト指向言語
 - 関数型言語
 - 宣言的で非アルゴリズム的な言語
 - スクリプト言語
- プログラミング方法論への規模の影響

学習の目標:

1. プログラミング言語の発展の歴史を要約し、現在使われているパラダイムがどんな歴史的経緯から生まれたかを述べる。
2. この科目がカバーしているプログラミングパラダイムのそれぞれについて、少なくとも一つの

特徴を挙げる。

- 異なるパラダイムの間でのトレードオフを評価する。この際、メモリ効率、時間効率（コンピュータとプログラマの両方について）、安全性、表現力などの問題を考慮する。
- 小規模プログラミングと大規模プログラミングの違いを理解する。

PL2 仮想計算機 [コア]

最低履修時間:1 時間

トピックス:

仮想計算機
仮想計算機の階層
中間言語
他人の計算機でコードを実行することから生じるセキュリティ問題

学習の目標:

- 仮想計算機において抽象化の重要性と能力を示す。
- コンパイルプロセスにおける中間言語の利点を説明する。
- 性能と可搬性のトレードオフを評価する。
- 実行可能なプログラムがディスクファイルやメモリをアクセスすることによって、いかにコンピュータシステムのセキュリティを破壊しうるかを説明する。

PL3 言語翻訳入門 [コア]

最低履修時間:2 時間

トピックス:

インタプリタとコンパイラの比較
言語翻訳の各フェーズ

- 字句解析
- 構文解析
- コード生成
- 最適化

言語翻訳における機械依存な部分と機械独立な部分

学習の目標:

- コンパイラ方式の実行モデルとインタプリタ方式の実行モデルを比較し、それぞれの長所を把握する。
- ソースコードから実行可能コードまでの言語翻訳の各フェーズ、およびこれらのフェーズで生成されるファイルについて述べる。

3. 機械依存翻訳と機械独立翻訳の違いを説明し、翻訳プロセスのどこでこれらの違いが明らかになるかを述べる。

PL4 宣言と型 [コア]

最低履修時間:3 時間

トピックス:

値の集合と操作の集合で規定される型の概念

宣言モデル

- 結合
- 可視性
- 有効範囲
- 生存期間

型チェックの概要

ごみ集め

学習の目標:

1. 特に大規模プログラミングに関して宣言モデルの価値を説明する。
2. アドレス, 値, 有効範囲, 持続性, サイズなどの変数の特性を確認し, 記述する。
3. 型の不一致について論じる。
4. 結合, 可視性, 有効範囲, 生存期間のそれぞれについて, いろいろな管理の方法を示す。
5. 抽象化および安全性を提供する際の型および型チェックの重要性について述べる。
6. 生存期間管理におけるトレードオフ (参照数カウント対ごみ集め) を評価する。

PL5 抽象化メカニズム [コア]

最低履修時間:3 時間

トピックス:

抽象化メカニズムとしての手続き, 関数, 反復子

引数のメカニズム (参照渡しと値渡し)

起動レコードと記憶管理

型パラメータおよびパラメータ化された型

プログラミング言語におけるモジュール

学習の目標:

1. 再利用可能なソフトウェア部品を作る際, いかに抽象化メカニズムが有用であるかを明らかにする。
2. 引数受渡しの二つの方法, 値渡しと参照渡しの違いを述べる。

3. 抽象化の重要性を, 特に大規模プログラミングに関して述べる。
4. プログラムモジュールとそのデータを管理するために, コンピュータシステムがどのように起動レコードを使用するかを示す。

PL6 オブジェクト指向プログラミング [コア]

最低履修時間:10 時間

トピックス:

オブジェクト指向設計
カプセル化と情報隠蔽
動作とその実装の分離
クラスとサブクラス
継承 (オーバーライド, 動的ディスパッチ)
多相性 (サブタイプ多相性と継承)
クラス階層
コレクションクラスと反復プロトコル
オブジェクトとメソッドテーブルの内部表現

学習の目標:

1. オブジェクト指向設計の考え方, および, カプセル化, 抽象化, 継承, 多相性の概念を説明する。
2. オブジェクト指向プログラミング言語によって簡単なプログラムを設計, 実装, テスト, デバッグする。
3. クラスのメカニズムがカプセル化や情報隠蔽にとって有用である理由について述べる。
4. クラス階層と継承を使用して, オブジェクト間の「is-a」関係の具体化を設計, 実装, テストする。
5. オブジェクト指向言語における多重定義とオーバーライドの概念を比較対照する。
6. クラスの静的構造とクラスのインスタンスの動的構造との関係を説明する。
7. 反復子がコンテナの要素にどのようにアクセスするかを示す。

PL7 関数型プログラミング [選択]

トピックス:

関数型言語の概要と目標
リスト, 自然数, 木を初めとする再帰的に定義されたデータ上の再帰
実際の使用法 (分割統治法によるデバッグ, データ構造の持続性)
関数データ構造に関するならし効率
閉包, データとしての関数の使用 (無限集合, ストリーム)

学習の目標:

1. 関数型プログラミングパラダイムの長所と短所を要約する。
2. 関数型パラダイムを使って, プログラムを設計し, コーディング, テスト, デバッグする。
3. データとしての関数の使用法を説明する。閉包の概念も含める。

PL8 言語翻訳システム [選択]

トピックス:

字句解析における正規表現の適用
 構文解析 (具象構文, 抽象構文, 抽象構文木)
 文脈自由文法へのテーブル駆動および再帰下降型構文解析の適用
 記号表の管理
 木の巡回によるコード生成
 アーキテクチャに特化した操作: 命令選択とレジスタ割当て
 最適化技法
 翻訳プロセスを支援するツールの使用およびその利点
 プログラムライブラリと分離コンパイル
 構文主導型ツールの構築

学習の目標:

1. 言語翻訳プログラムのステップと, それぞれで使用するアルゴリズムを示す。
2. 有限状態オートマトン, プッシュダウンオートマトンなどの内在する形式モデルを認識し, それらが言語定義にどう関係しているかを正規表現や文法を通じて理解する。
3. 最適化の有効性を論じる。
4. 分離コンパイルやプログラムライブラリの存在がコンパイルプロセスにどう影響するかを説明する。

PL9 型システム [選択]

トピックス:

値および操作の集合としてのデータ型
 データ型

- 基本型
- 直積型と直和型
- 代数型
- 再帰的な型
- 矢印 (関数) 型
- パラメータ付きの型

 型チェックモデル
 ユーザ定義型の意味論モデル

- 型の省略

- 抽象データ型
 - 型の同等性
- パラメータ多相性
サブタイプ多相性
型チェックのアルゴリズム

学習の目標:

1. 型の考え方を形式化する。
2. 基本データ型のそれぞれについて述べる。
3. 抽象データ型の概念を明らかにする。
4. 抽象化や安全のための型の重要性を認識する。
5. 静的な型と動的な型を区別する。
6. 型宣言と型推論を区別する。
7. 型の観点から言語を評価する。

PL10 プログラミング言語の意味論 [選択]

トピックス:

- 非形式的な意味論
形式的意味論の概観
表示の意味論
公理の意味論
操作の意味論

学習の目標:

1. 形式的意味論の重要性を認識する。
2. 形式的意味論と非形式的意味論を区別する。
3. 形式的意味論への各種のアプローチについて述べる。
4. 形式的意味論への各種のアプローチを評価する。

PL11 プログラミング言語の設計 [選択]

トピックス:

- 言語設計の一般原則
設計目標
型の体系
データ構造のモデル
制御構造のモデル
抽象化メカニズム

学習の目標:

1. 言語設計, 言語使用, 翻訳プロセスに対する異なった型の体系の影響を評価する。
 2. ユーザ定義機能の作成における様々な抽象化メカニズムの役割を説明する。
-

ヒューマンコンピュータインタラクション (HC)

- HC1 . ヒューマンコンピュータインタラクションの基礎 [コア]
- HC2 . 簡単なグラフィカル・ユーザインタフェースの構築 [コア]
- HC3 . 人間中心のソフトウェア評価 [選択]
- HC4 . 人間中心のソフトウェア開発 [選択]
- HC5 . グラフィカル・ユーザインタフェースの設計 [選択]
- HC6 . グラフィカル・ユーザインタフェースのプログラミング [選択]
- HC7 . マルチメディアシステムの HCI 的側面 [選択]
- HC8 . 協同作業とコミュニケーションの HCI 的側面 [選択]

ここに挙げた科目リストは、コンピュータ科学専攻学生のためのヒューマンコンピュータインタラクション入門を意図している。ここでの重点は、対話型オブジェクトに対する人間行動の理解、人間中心アプローチによる対話型ソフトウェアの開発および評価方法の理解、ならびに、対話型ソフトウェアの様々なタイプにおいて生じる HCI 設計上の問題に関する一般的知識である。ユニット HC1 (ヒューマンコンピュータインタラクションの基礎) およびユニット HC2 (簡単なグラフィカル・ユーザインタフェースの構築) は、すべての専攻学生に求められるが、可能であれば入門コースの一部として実施すべきである。その他のユニットは、多くの場合、3 年生または 4 年生レベルの一つか二つの選択科目に統合される。

HC1 ヒューマンコンピュータインタラクションの基礎 [コア]

最低履修時間:6 時間

トピックス:

- 動機づけ : なぜ人間を考慮するのか
- HCI の背景 (ツール, ウェブハイパーメディア, 通信)
- 人間中心の開発および評価
- 人間行動モデル : 知覚, 動作, 認知
- 人間行動モデル : 文化, コミュニケーション, 組織
- 人間の多様性への対応
- よい設計およびよい設計者の原理 : エンジニアリングトレードオフ
- ユーザビリティテスト入門

学習の目標:

1. 人間中心のソフトウェア開発を行う理由について論じる。
2. 心理的および社会的相互作用の基礎科学を要約する。
3. 仮説および実験結果の役割と、相関関係の役割との違いを区別する。
4. 人間とソフトウェアの対話を分析するための概念的語彙を構築する : アフォーダンス, 概念モデル, フィードバックなど。
5. 特定のアイコン, 記号, 単語, または色が, (a) 二つの異なる人間文化や, (b) 一つの文化お

よびその下位文化においてもつ異なった解釈を区別する。

6. 人間の多様性を考慮するという点から見て、コンピュータシステムまたはアプリケーションの設計はどのように成功または失敗するのであろうか。
7. 現存するソフトウェアアプリケーションのための、簡単なユーザビリティテストを作成し、実施する。

HC2 簡単なグラフィカル・ユーザインタフェースの構築 [コア]

最低履修時間:2 時間

トピックス:

グラフィカル・ユーザインタフェース (GUI) の原理
GUI ツールキット

学習の目標:

1. 効果的なグラフィカル・ユーザインタフェース設計のためのいくつかの基本原理を確認する。
2. GUI ツールキットを使用して、グラフィカル・ユーザインタフェースを伴った簡単なアプリケーションを作成する。
3. グラフィカル・ユーザインタフェースの構造に対する基本的な設計原理の影響を例証する。
4. 各インスタンスに対して簡単なユーザビリティテストを行い、結果を比較する。

HC3 人間中心のソフトウェア評価 [選択]

トピックス:

評価目標の設定

ユーザの参加なしで行う評価

- ウォークスルー
- キーストロークレベルモデル (KLM)
- ガイドライン
- 標準

ユーザが参加して行う評価

- ユーザビリティテスト
- インタビュー
- アンケート調査
- 実験

学習の目標:

1. 評価基準について論じる: 学習, タスク時間と完了度, 容認可能性。
2. ウォークスルーおよびキーストロークレベルモデル (KLM) 分析を実施する。
3. 主要なガイドラインおよび標準を要約する。

4. ユーザビリティテスト, インタビュー, およびアンケート調査を実施する。
5. ユーザビリティテストと制御環境下における実験とを比較する。
6. 現存する対話型システムを, 人間中心の基準およびユーザビリティテストによって評価する。

HC4 人間中心のソフトウェア開発 [選択]

トピックス:

アプローチ, 特性, およびプロセスの概観

機能性とユーザビリティ: タスク分析, インタビュー, アンケート調査

対話と表現の仕様記述

プロトタイピング技法とツール

- 紙を使ったストーリーボード
- 継承および動的ディスプレイ
- プロトタイピング言語および GUI ビルダ

学習の目標:

1. 人間中心開発の基本的な種類および特徴を説明する。
2. 人間中心開発を, 伝統的なソフトウェア工学手法と比較する。
3. 機能要求とユーザビリティ要求をそれぞれ 3 件ずつ記述する。
4. 対話型オブジェクトの仕様を, 遷移ネットワーク, オブジェクト指向設計, またはシナリオ記述を用いて記述する。
5. 紙の上での開発とソフトウェアプロトタイプによる開発の長所, 短所について論じる。

HC5 グラフィカル・ユーザインタフェースの設計 [選択]

トピックス:

対話スタイルおよび対話技法の選択

共通ウィジェットの HCI 的側面

画面設計の HCI 的側面: レイアウト, 色, フォント, ラベリング

ヒューマンエラーの扱い

単純な画面設計を超えたもの: 視覚化, 表現, メタファ

マルチモーダルな対話: グラフィックス, 音, 触覚

3次元対話と仮想現実

学習の目標:

1. 一般的な対話スタイルを要約する。
2. 次の各々についてよい設計原理を説明する: 一般のウィジェット, 画面表示の順序, 簡単なエラートラップダイアログ, ユーザマニュアル。
3. HC3 および HC4 で扱った概念知識を使って, 簡単な 2次元 GUI を設計し, プロトタイプを

作成し, 評価する。

4. 2次元対話から3次元へと移行する際の課題について論じる。

HC6 グラフィカル・ユーザインタフェースのプログラミング [選択]

トピックス:

ユーザインタフェース管理システム (UIMS), 対話の独立と分析のレベル, Seeheim モデル
 ウィジェットクラス
 イベント管理とユーザ対話
 ジオメトリ管理
 GUIビルダと UI プログラミング環境
 クロスプラットフォーム設計

学習の目標:

1. UIMS の責任とアプリケーションの責任を区別する。
2. UI のためのカーネルベースモデルとクライアントサーバモデルを区別する。
3. UI のイベント駆動型パラダイムと, より伝統的な手続き型制御とを比較する。
4. ウィジェットの集合体および制約に基づくジオメトリ管理について説明する。
5. コールバックについて説明し, GUIビルダにおけるその役割を述べる。
6. クロスプラットフォーム UI 設計に共通する差異を少なくとも三つ確認する。
7. 様々なプラットフォームの UI に見られる共通点をできるだけ多く確認する。

HC7 マルチメディアシステムの HCI 的側面 [選択]

トピックス:

情報の分類とアーキテクチャ : 階層, ハイパーメディア
 情報検索と人間の行動

- ウェブ検索
- データベース問合わせ言語のユーザビリティ
- グラフィックス
- 音

 マルチメディア情報システムの HCI 設計
 音声認識と自然言語処理
 情報機器とモバイルコンピューティング

学習の目標:

1. 情報検索がトランザクション処理とどのように異なるのかを論じる。
2. 情報の構成がどのように検索を支えているのか, その仕組みを説明する。
3. データベース問合わせ言語に関する主要なユーザビリティ問題を述べる。
4. 音声技術の現状, さらに自然言語処理一般の現状について説明する。

5. HC4, HC5, および HC7 で扱った概念知識を使って, 簡単なマルチメディア情報システムを設計し, プロトタイプを作成し, 評価する。

HC8 協同作業とコミュニケーションの HCI 的側面 [選択]

トピックス:

専門的タスクを支援するグループウェア : ドキュメント作成, マルチプレイヤーゲーム
非同期グループコミュニケーション : 電子メール, 掲示板
同期グループコミュニケーション : チャットルーム, 電子会議
オンラインコミュニティ : MUD/MOO
ソフトウェアキャラクタと知的エージェント

学習の目標:

1. 個別対話における HCI 問題を, グループ対話における問題と比較する。
 2. 協同作業ソフトウェアによって生じる社会的な諸問題について論じる。
 3. 人間の意図を具体的に表現するソフトウェアにおける HCI 問題について論じる。
 4. 同期コミュニケーションと非同期コミュニケーションの差異を述べる。
 5. HC4, HC5, および HC8 で扱った概念知識を使って, 簡単なグループウェアまたはグループコミュニケーションアプリケーションを設計し, プロトタイプを作成し, 評価する。
 6. 対面での対話形式と, 仲介ソフトウェア環境を通じたその他の対話形式が混在したチームプロジェクトに参加する。
 7. 対面での協同作業とソフトウェアを仲介した協同作業の間の類似点および相違点を述べる。
-

グラフィックスとビジュアル・コンピューティング (GV)

- GV1 . グラフィックスにおける基礎技術 [コア]
- GV2 . グラフィック・システム [コア]
- GV3 . グラフィック・コミュニケーション [選択]
- GV4 . 幾何モデリング [選択]
- GV5 . レンダリングの基本 [選択]
- GV6 . 高度なレンダリング [選択]
- GV7 . 高度な技術 [選択]
- GV8 . コンピュータ・アニメーション [選択]
- GV9 . 視覚化 [選択]
- GV10 . 仮想現実 [選択]
- GV11 . コンピュータ・ビジョン [選択]

グラフィックスとビジュアル・コンピューティング (GV) によって包含される領域は次のような四つの相互に関連した分野に分けられる。

コンピュータ・グラフィックス：コンピュータ・グラフィックスは、コンピュータの処理を介して生成され、表現されるような画像を利用した情報コミュニケーションの芸術および科学である。これは次の四つの項目を必要とする。(a) 画像の創作と見方を支援する様々な方法において、情報を表現するようなモデルのデザインと構成、(b) モデルやビューと対話する人との間に介在する入出力の機材のデザインや技術、(c) モデルをレンダリングする技術の創造、(d) 保存管理される画像の設計方法。コンピュータ・グラフィックスの目的は、人の視覚を他の感覚とともに、理解のために利用することにある。

視覚化：視覚化の分野では、科学的（コンピュータ科学や医療科学）またはより抽象的なデータ集合に内在する相互に関連づけられた構造や関係を判定し、表現する方法を探る。表現の主目的は、理解を促進するような形で、データ集合による情報を伝えることである。視覚化の現在ある技術は人間の視覚的な能力を利用しているが、音や触覚なども含めた他の知覚も、情報の発見過程を支援するために考慮されつつある。

仮想現実：仮想現実 (VR) は、コンピュータ・グラフィックスや場合によっては他の知覚を使って生成された 3 次元の環境をユーザに経験させることを可能とする。そしてその環境は、コンピュータが生成した世界と人間の間の相互作用を強化するためにある。

コンピュータ・ビジョン：コンピュータ・ビジョン (CV) の目的は、一つあるいは複数の 2 次元画像から、3 次元の世界の特徴や構造を推論することにある。コンピュータ・ビジョンの理解と実践は、コンピュータ科学の核となる概念に依存しているだけでなく、さらに物理、数学、心理学の原理にも強く関係している。

GV1 グラフィックスにおける基礎技術 [コア]

最低履修時間:2 時間

トピックス:

グラフィックスソフトウェアの階層
グラフィックス API の使用
単純なカラーモデル (RGB, HSB, CMYK)
同次座標
アフィン変換 (スケーリング, 回転, 平行移動)
ビュー変換
クリッピング

学習の目標:

1. グラフィックスソフトウェアの異なるレベルの能力を区別する。そしてそれぞれがどんな場合に適切であるかを評価する。
2. 標準のグラフィックス API を用いて画像を生成する。
3. 標準の API によって与えられる機能を使って, スケーリング, 回転, 平行移動などの基本的な座標変換を表現する。
4. 座標変換やクリッピング操作を単純な 2 次元画像に対して実行するような簡単な手続きを実装する。
5. 3 次元座標系について議論する。また, 2 次元の座標変換演算を 3 次元に拡張するための変更点について議論する。

GV2 グラフィック・システム [コア]

最低履修時間:1 時間

トピックス:

ラスターおよびベクトル・グラフィックス・デバイス
ビデオ・ディスプレイ・システム
物理的入力機器と論理的入力機器
グラフィックなシステムの開発者が直面する問題

学習の目標:

1. 与えられた応用に対してグラフィックス・アーキテクチャの適切性を評価する。
2. 様々な入力機器の機能を説明する。
3. ラスター・グラフィックスとベクトル・グラフィックスの間の技術の違いを比較対照する。
4. 現行のハードウェアやソフトウェアを使って, グラフィックスを作り出し, 表示する。
5. グラフィックスを作り出し, 表示するための, これから現れてくるハードウェアやソフトウェアの拡張された能力に関して議論する。

GV3 グラフィック・コミュニケーション [選択]

トピックス:

色や色の相互作用による心理的な力
 視覚障害者のための色の変更
 異なる色への文化的な意味付け
 特定の視聴者に対する効果的な擬似色のパレットを画像に適用する方法
 効果的な理解のためのビューの構造化
 効果的なビデオや印刷された画像への画像の変更
 色や他の視覚的なデータへの情報の対応づけにおける慣習の利用
 文脈や背景となる情報を表現するための画像内でのテキストの使用
 グラフィック的な操作におけるユーザへの視覚的なフィードバック

学習の目標:

1. 色および擬似色の使用の意義を説明する。
2. 効果的なビデオや印刷された画像を作成できる能力があることを示す。
3. グラフィックスを用いたコミュニケーションの効果的な例と効果がない例を識別する。
4. 色, 記号, テキスト, 映像などを適切に用いて, 効果的なグラフィック・コミュニケーションの作品を創作する。
5. 同一の内容を伝える二つの効果的な作品を創作する。一方は印刷媒体向けにデザインされており, 他方はオンライン表示用にデザインされていること。
6. 印刷媒体とオンライン媒体に対するデザインの指針における違いを議論する。

GV4 幾何モデリング [選択]**トピックス:**

3次元物体のポリゴン表現
 パラメトリック多項式曲線および曲面
 CSG (ソリッド図形による構成的な) 表現
 曲線や曲面の陰関数形式
 空間の細分割技術
 手続きモデル
 変形可能モデル
 細分割曲面
 複数の解像度が混在するモデリング
 再構成

学習の目標:

1. 表面のテッセレーションによって簡単な多面体を作成する。
2. 立方体や2次曲面のような単純なプリミティブ図形から CSG モデルを構築する。
3. 陰関数で表現された曲面からメッシュ表現を生成する。

4. 手続き的な手法でフラクタル・モデルやフラクタル地形を生成する。
5. レーザ・スキャンによって得られた地点データからメッシュを生成する。

GV5 レンダリングの基本 [選択]

トピックス:

直線生成アルゴリズム (ブレゼンハム)
フォントの生成 (アウトライン対ビットマップ)
光源および質感特性
環境光, 拡散反射光, 鏡面反射光
フォンの反射モデル
ポリゴン面のレンダリング (フラット・シェーディング, グローブ・シェーディング, フォン・シェーディング)
テキスト・マッピング, バンプ・テキスト, 環境マッピング
レイ・トレーシング入門
画像合成, 標本化技術, アンチエイリアシング

学習の目標:

1. ピクセル方式のディスプレイ上に線をレンダリングするための, ブレゼンハムのアルゴリズムの動作を説明する。
2. このユニットに挙げた技術のそれぞれに関して, その概念と応用について説明する。
3. 標準的な API を利用して画像を作成することで, これらの技術のそれぞれを紹介する。
4. あるグラフィック画像がどのように作られたか記述する。

GV6 高度なレンダリング [選択]

トピックス:

輸送方程式
レイ・トレーシングのアルゴリズム
フォトン・トレーシング
大局照明計算のためのラジオシティ法, 形態係数
大局照明への効率的なアプローチ
大局照明のためのモンテカルロ法
画像に基づくレンダリング, パノラマ画像, プレノプティック関数モデリング
複雑な自然現象のレンダリング
非写実レンダリング

学習の目標:

1. いくつかの輸送方程式について (すべての包括的な効果を含めて) 詳細に記述する。
2. ラジオシティを計算する効率的なアルゴリズムについて記述し, 正確さとアルゴリズムの実

行速度とのトレードオフを説明する。

3. メッシュ化方式の影響について記述する。
4. 画像に基づくレンダリング技術, 空間光線, および関連する事項について説明する。

GV7 高度な技術 [選択]

トピックス:

色の量子化

2D プリミティブのスキャン変換, 前進差分

曲面のテッセレーション

隠面消去手法

Z バッファおよびフレーム・バッファ, カラー・チャンネル (透明度のチャンネル)

高度な幾何モデリング技術

学習の目標:

1. このユニットに挙げられた技術を記述する。
2. 特定の画像を生成するために使用されたグラフィック技術をどのように推定するか説明する。
3. 単純なグラフィック・システムを用いて, 個々のピクセルのレベルで, 指定されたいずれかのグラフィック技術を実現する。
4. 一般的なアニメーション・ソフトウェアを使って, メタボールやスケルトンを使った簡単な有機形状を構築する。

GV8 コンピュータ・アニメーション [選択]

トピックス:

キーフレーム・アニメーション

カメラ・アニメーション

スクリプティング・システム

関節のある構造のアニメーション : インバース・キネマティクス

モーション・キャプチャ

手続き的なアニメーション

デフォーム

学習の目標:

1. 中間地点の位置と方向を生成するためのスプライン内挿法を説明する。
2. モーション・キャプチャのためのいくつかの技術を対比して比較する。
3. 花火のような単純なアニメーションを生成するために, 一般的なアニメーション・ソフトウェアにおいて, パーティクルの機能を使用する。
4. 多様な変形を作り出すための自由形状のデフォームを使用する。

GV9 視覚化 [選択]

トピックス:

視覚化のための基本的なビューおよび質問関数
ベクトル場, テンソル, 流れデータの視覚化
スカラー場あるいは高度のある場の視覚化 (立方体適合手法による等表面)
直接ボリューム・レンダリング (レイ・キャスティング, 移動関数, セグメンテーション, ハードウェア)
情報の可視化 (投影法, 平行座標法)

学習の目標:

1. スカラーとベクトルの視覚化の背景にある基本的なアルゴリズムを記述する。
2. 正確さと実行速度の点からアルゴリズムのトレードオフを評価する。
3. 視覚化操作の効果を説明するために信号処理や数値解析から適当な理論を選んで使用する。
4. 視覚的な表現やユーザとの視覚的な対話手法が, 探究に及ぼす影響力を記述する。

GV10 仮想現実 [選択]

トピックス:

ステレオスコピック・ディスプレイ
力のフィードバックのシミュレーション, 触覚の入出力機器
ビューワ追跡
衝突感知
視界の計算
時間制限のあるレンダリング, 複数の詳細レベル (LOD)
画像に基づく VR システム
分散 VR, コンピュータ・ネットワーク上の共同作業
対話的なモデリング
ユーザインタフェースの問題点
医療, シミュレーション, トレーニングにおける応用

学習の目標:

1. ステレオスコピック・ビューを合成するためにコンピュータ・グラフィックス・システムによって実現される光学モデルを記述する。
2. 異なるビューワ追跡技術の原理を記述する。
3. 凸多面体用の効率的な衝突感知アルゴリズムの原理を説明する。
4. 幾何ベースと画像ベースの仮想現実の違いを評価する。
5. ネットワーク化された環境におけるユーザ・アクションの同期とデータの一貫性の問題点について記述する。

6. 特定の応用のための VR システムのインタフェース, ハードウェア, ソフトウェア構成についての基本的な要求事項を決定する。

GV11 コンピュータ・ビジョン [選択]

トピックス:

画像獲得

デジタル画像とその特性

画像のプリプロセス

セグメンテーション (スレッシュホルディング, エッジに基づく, 領域に基づく)

形状表現と物体認識

モーションの解析

ケーススタディ (物体認識, 物体追跡)

学習の目標:

1. 画像構成処理について説明する。
 2. 二つ以上のカメラや, ステレオ・ビジョンの有用性を説明する。
 3. 多様な分割の方法について, それらの特徴, 差異, 長所, 短所を含めて説明する。
 4. 輪郭および領域を基本とした形状表現に基づいた物体認識を記述する。
 5. 差分によるモーション解析法を説明する。
 6. 物体追跡法の各種方法の違いを述べる。
-

インテリジェントシステム (IS)

- IS1 . インテリジェントシステムの基本的問題 [コア]
- IS2 . 探索および制約充足 [コア]
- IS3 . 知識表現および推論 [コア]
- IS4 . 高度な探索 [選択]
- IS5 . 高度な知識表現と推論 [選択]
- IS6 . エージェント [選択]
- IS7 . 自然言語処理 [選択]
- IS8 . 機械学習とニューラルネット [選択]
- IS9 . AI プラニングシステム [選択]
- IS10 . ロボット工学 [選択]

人工知能 (AI) は、自律エージェントの設計および分析に関する分野である。自律エージェントは、センサおよびアクチュエータをもつソフトウェアシステム、あるいは物理的機械であって、例えばロボットあるいは無人の宇宙船の中などに組み込まれている。インテリジェント (知的) システムは、それが置かれた環境を知覚し、定められた作業のために合理的に行動し、他のエージェントおよび人間と相互に作用しなければならない。

これらの機能については、コンピュータビジョン、プログラミングと実行、ロボット工学、マルチエージェントシステム、音声認識、および自然言語認識といった科目によってカバーされる。これらの科目は、広範囲な一般的あるいは専門的な知識表現と推論機構、問題解決と探索アルゴリズム、および機械学習技術に依存する。

さらに人工知能は、他の手段では解決が困難または実際的でない問題への解決ツールを提供する。これには、発見的探索 (heuristic search) およびプログラミングアルゴリズム、知識表現および推論のための形式化、機械学習技術、音声認識、言語理解、コンピュータビジョン、ロボット工学のような認知・動作問題に適用できる手法が含まれている。学生は、与えられた問題に対して、どんな場合に人工知能アプローチを適用すべきかを決定し、適切な人工知能手法を選択して実現する能力が必要とされる。

IS1 インテリジェントシステムの基本的問題 [コア]

最低履修時間:1 時間

トピックス:

人工知能の歴史

哲学的質問

- チューリング・テスト
- サールの「中国語の部屋」の思考実験
- AI における倫理問題

基本的定義

- 最適推論と人間的推論
- 最適行動と人間的行動

世界のモデル化

発見的知識 (ヒューリスティックス) の役割

学習の目標:

1. チューリング・テスト, およびサールの反論 (中国語の部屋) について述べる。
2. 最適推論と人間的推論の概念を区別する。
3. 最適行動と人間的行動の概念を区別する。
4. 世界モデルに依存するインテリジェントシステムの例をいくつか挙げる。
5. 発見的知識の役割を述べ, 最適性と効率の間のトレードオフの必要性を示す。

IS2 探索および制約充足 [コア]

最低履修時間: 5 時間

トピックス:

問題空間

力づく探索

- 幅優先
- 深さ優先
- 反復深化による深さ優先

最良優先探索

- 一般的な最良優先
- ダイクストラのアルゴリズム
- A*アルゴリズム
- A*アルゴリズムの最適性保証

2 プレーヤゲーム

- ミニマックス法
- α - β 枝刈り

制約充足 (バックトラック法および局所探索法)

学習の目標:

1. 英語で表された課題を次の四つを用いて表すことにより, 効率的な問題空間を定式化する: 状態, 操作, 初期状態, 目標状態。
2. 組合せ的爆発の問題とその結果を述べる。
3. 問題に対して適切な力づく探索アルゴリズムを選び, それを実現して, その時間計算量および領域計算量を調べる。
4. 問題に対して適切な発見的探索アルゴリズムを選び, それに必要な発見的評価関数を設計することによって実現する。

5. どのような条件が成立すれば, 発見的アルゴリズムによって最適解が得られると保証されるか, について述べる。
6. 2プレーヤゲームを取り上げ, α - β 枝刈りを含んだミニマックス法を実現する。
7. 英語で記述された問題を制約充足問題として定式化し, 時間順バックトラックアルゴリズムを用いて実現する。

IS3 知識表現および推論 [コア]

最低履修時間:4 時間

トピックス:

命題論理および述語論理の復習
導出, 定理証明
非単調推論
確率的推論
ベイズの定理

学習の目標:

1. 定理証明のための導出原理の操作を説明する。
2. 単調推論と非単調推論の違いを説明する。
3. 確率的推論の長所と短所について論じる。
4. ベイズの定理を適用して条件付き確率を求める。

IS4 高度な探索 [選択]

トピックス:

遺伝的アルゴリズム
焼きなまし法 (simulated annealing)
局所探索

学習の目標:

1. 遺伝的アルゴリズムとは何であることを説明し, その有効性を従来の問題解決および探索の技術と対照する。
2. 焼きなまし法が探索の複雑さを減少させるための利用方法を述べ, その操作を従来 of 探索技術と対照する。
3. 局所探索法を従来 of 問題領域に適用する。

IS5 高度な知識表現と推論 [選択]

トピックス:

構造化表現

- フレームとオブジェクト
- 記述論理
- 継承システム

非単調推論

- 非古典論理
- デフォルト推論
- 信念変更
- 選好論理
- 知識源の統合
- 矛盾する信念の集合

行為および変化に関する推論

- 状況計算
- 事象計算
- 派生する問題

時制推論と空間推論**不確実性**

- 確率的推論
- ベイズネット
- ファジィ集合および可能性理論
- 決定理論

対話のための知識表現, 定性表現

学習の目標:

1. 知識の構造化表現に使用される最も一般的なモデルを, 長所と短所を強調しつつ, 比較対照する。
2. 非単調推論の構成要素を挙げ, 信念システムのための表現機構としての非単調推論の有用性を述べる。
3. 行為と変化の問題に状況計算や事象計算を適用する。
4. 時制推論と空間推論の区別を述べ, どのように相互に関係しているかを説明する。
5. 不確実性を表現するための各種基本技術を述べ, 対照する。
6. 対話と定性表現のための各種基本技術を説明し, 対照する。

IS6 エージェント [選択]**トピックス:****エージェントの定義**

成功した応用事例およびエージェントに基づく最新のシステム

エージェントアーキテクチャ

- 簡単な即応エージェント
- 即応プランナ

- 多層アーキテクチャ
- アーキテクチャの例と応用

エージェント理論

- コミットメント
- 意図
- 決定理論的エージェント
- マルコフ決定過程 (MDP)

ソフトウェアエージェント, 個人支援, 情報アクセス

- 協調エージェント
- 情報収集エージェント

信念をもつことのできるエージェント (合成特徴, エージェントにおける感情のモデル化)

学習エージェント

マルチエージェントシステム

- 経済的動機によるマルチエージェントシステム
- 協調エージェント
- エージェントチーム
- エージェントモデリング
- マルチエージェント学習

ロボットエージェント入門

モバイルエージェント

学習の目標:

1. エージェントは他の種類のインテリジェントシステムとどのように異なるかを説明する。
2. 標準的なエージェントアーキテクチャを調べ, 対照する。
3. ソフトウェアエージェント, 個人支援, 信念をもつことのできるエージェントなどの問題領域に対するエージェント理論の応用を述べる。
4. 学習するエージェントと学習しないエージェントの違いを述べる。
5. 適当な例を使って, マルチエージェントシステムがエージェント間のやりとりをどのように支援するかについて例示する。
6. ロボットエージェントとモバイルエージェントについて説明し, 対照する。

IS7 自然言語処理 [選択]

トピックス:

決定的文法と確率的文法
構文解析アルゴリズム
コーパスに基づく手法
情報検索
言語翻訳
音声認識

学習の目標:

1. 決定的文法および確率的文法の定義を述べ、対照する。それぞれが適切であるような場合の例を示す。
2. 自然言語を構文解析するための従来のアルゴリズムを述べる。
3. 確立したコーパスの利用の必要性を論じる。
4. コーパスに基づくアプローチにおける格納手続きや探索手続きの例を示す。
5. 情報検索, 言語翻訳, および音声認識の技術の違いを明確にする。

IS8 機械学習とニューラルネット [選択]

トピックス:

機械学習の定義および例
教師あり学習
決定木学習
ニューラルネット学習
信念ネットワーク学習
最近傍アルゴリズム
学習理論
過学習問題
教師なし学習
強化学習

学習の目標:

1. 次の三つの主要学習方式の違いを説明する: 教師あり学習, 強化学習, 教師なし学習。
2. 教師あり学習, 強化学習, および教師なし学習のための簡単なアルゴリズムを実現する。
3. 特定の問題領域に対して, 三つの学習スタイルのうちのいずれが適しているかを判断する。
4. 決定木, ニューラルネット, 信念ネットワークの三つの方式を比較対照し, それぞれがどのようなときに優れているかについて, 例を挙げながら示す。
5. 決定木, ニューラルネット, 信念ネットワークから適切なものを選んで, 簡単な学習システムを実現する。
6. 学習理論における最新動向を調査し, その到達点と問題点を示す。
7. 学習理論における最近傍アルゴリズムとその位置づけを説明する。
8. 過学習問題を検出・制御するための技術を説明する。

IS9 AI プラニングシステム [選択]

トピックス:

プラニングシステムの定義および例
探索としてのプラニング
操作に基づくプラニング

命題プランニング

プランニングシステムの拡張 (事例ベース, 学習, および確率的システム)

静的世界プランニングシステム

プランニングと実行

プランニングとロボット工学

学習の目標:

1. プランニングシステムの概念の定義を述べる。
2. プランニングシステムが従来の探索技法とどのように異なるかを説明する。
3. 探索としてのプランニング, 操作に基づくプランニング, および命題プランニングの違いについて, 最も適切な領域の例を挙げてそれぞれ説明する。
4. 事例に基づくプランニング, 学習プランニング, および確率的プランニングの各技術について, 定義を述べ例を示す。
5. 静的世界プランニングシステムと動的実行を必要とするシステムを比較対照する。
6. ロボット工学における動的プランニングの効果を説明する。

IS10 ロボット工学 [選択]

トピックス:

概要

- 最新のロボットシステム
- プランニングと即応制御
- 制御における不確実性
- センサ感知
- 世界モデル

構成空間

プランニング

センサ感知

ロボットプログラミング

ナビゲーションおよび制御

学習の目標:

1. 今日の最新ロボットシステムの将来性および限界を概観する。
2. 2次元ロボットおよび複雑な多角形のための構成空間アルゴリズムを実現する。
3. 簡単な運動のプランニングアルゴリズムを実現する。
4. センサに関連する不確実性とその不確実性に対処する方法を説明する。
5. 簡単な制御アーキテクチャを設計する。
6. 未知の環境下における様々なナビゲーション戦略を述べ, それぞれの長所と短所を説明する。
7. 目印を用いる様々なナビゲーション戦略を述べ, それぞれの長所と短所を説明する。

情報管理 (IM)

- IM1 . 情報モデルとシステム [コア]
- IM2 . データベースシステム [コア]
- IM3 . データモデリング [コア]
- IM4 . 関係データベース [選択]
- IM5 . データベース問合わせ言語 [選択]
- IM6 . 関係データベース設計 [選択]
- IM7 . トランザクション処理 [選択]
- IM8 . 分散データベース [選択]
- IM9 . データベースの物理設計 [選択]
- IM10 . データマイニング [選択]
- IM11 . 情報格納と検索 [選択]
- IM12 . ハイパーテキストとハイパーメディア [選択]
- IM13 . マルチメディア情報とシステム [選択]
- IM14 . 電子図書館 [選択]

情報管理 (IM) は、コンピュータが使われるほとんどすべての分野で重要な役割を果たしている。この領域には次の項目が含まれる。情報に関する操作 (獲得, デジタル化, 表現, 構成, 変形, 提示), 記憶情報の効率的かつ効果的なアクセスおよび更新のためのアルゴリズム, データモデリングおよび抽象化, 物理的ファイル記憶装置技術。この領域はまた, 共用環境における情報安全性, 機密性, 整合性, および保護も含む。学生は, 概念データモデルおよび物理データモデルを開発することができ, 与えられた問題に対して, どのような情報管理方法および技術が適しているかを判断することができ, そして, 拡張性と利用可能性を含むすべての適切な制約条件を反映する, 適切な情報管理の解を選択し, 実現できることが必要である。

IM1 情報モデルとシステム [コア]

最低履修時間: 3 時間

トピックス:

- 情報システムの歴史および動機づけ
- 情報格納と検索 (IS と IR)
- 情報管理の応用
- 情報の獲得と表現
- 分析と索引付け
- 探索, 検索, 関連付け, ナビゲーション
- 情報機密性, 整合性, 安全性, 保持
- 拡張性, 効率, 有効性

学習の目標:

1. 情報というものをデータや知識と比較対照する。
2. 情報システムの初期のビジョンから近年の製品までの発展を、それぞれの能力、および将来の可能性を明確にして要約する。
3. 小規模ないし中規模程度の情報の応用プログラムが実際のユーザの情報ニーズを満たすかという点から批判し擁護する。
4. 情報機密性、整合性、安全性、および保持に関連した問題に対する様々な技術的解答を示す。
5. 効率性（処理能力、応答時間）および有効性（再現率、適合率）の測定基準を説明する。
6. 情報システムが個人用から汎用的なものにまで段階的に進むことを保証するアプローチを述べる。

IM2 データベースシステム [コア]

最低履修時間:3 時間

トピックス:

データベースシステムの歴史およびその動機づけ
データベースシステムの構成要素
DBMS（データベース管理システム）の機能
データベースアーキテクチャとデータ独立
データベース問合わせ言語の利用

学習の目標:

1. データベースアプローチを、データファイルを用いたプログラミングという伝統的なアプローチと区別する特徴を説明する。
2. データベースシステムの基本的目標、機能、モデル、構成要素、応用、および社会的影響について例を挙げて述べる。
3. データベースシステムの構成要素を示し、その利用例を述べる。
4. DBMS の主要な機能を述べ、データベースシステムにおけるその役割を述べる。
5. データベースシステムにおいてデータ独立の概念およびその重要性を説明する。
6. 問合わせ言語を使用してデータベースから情報を取り出す。

IM3 データモデリング [コア]

最低履修時間:4 時間

トピックス:

データモデリング
概念モデル（実体関連モデルおよび UML（統一モデリング言語）を含む）
オブジェクト指向モデル
関係データモデル

学習の目標:

1. データベースの構造を記述するために提供されている概念の種類に基づいて、データモデルを、概念データモデル、物理データモデル、表現データモデルに分類する。
2. 実体関連モデルと UML のモデリング概念および表記法を示し、データモデリングにおけるそれらの使用法を述べる。
3. オブジェクト同一性、型構成子、カプセル化、継承、多相性、およびバージョン管理など、オブジェクト指向モデルの主要な概念を示す。
4. 関係データモデルにおいて使用される基本的な用語の定義を述べる。
5. 関係データモデルの基本原理を述べる。
6. 関係データモデルのモデリング概念および表記法を例示する。

IM4 関係データベース [選択]**トピックス:**

概念スキーマの関係スキーマへのマッピング
 実体および参照整合性
 関係代数および関係論理

学習の目標:

1. 実体関連モデルを用いて開発された概念モデルから関係スキーマを作成する。
2. 実体整合性制約および参照整合性制約の概念（外部キー概念の定義を含む）を説明し、例示する。
3. 集合論に基づいた関係代数操作（和、積、差、直積）、および関係データベースのために開発された関係代数操作（選択、射影、結合、商）の利用方法を例示する。
4. 関係代数による問合わせを例示する。
5. タプル論理による問合わせを例示する。

IM5 データベース問合わせ言語 [選択]**トピックス:**

データベース言語の概要
 SQL（データ定義、問合わせ記述、更新言語、制約、整合性）
 問合わせ処理の最適化
 QBE および第 4 世代環境
 非手続き的問合わせの手続き型言語への埋め込み
 オブジェクト問合わせ言語入門

学習の目標:

1. SQL によって、キー制約、実体整合性制約、参照整合性制約を含む関係データベーススキ

- ーマを作成する。
- SQL におけるデータ定義を例示し, SQL の SELECT 命令を用いて, データベースから情報を検索する。
 - いくつかの問い合わせ処理方法を比較評価して, 最適の方法を選択する。
 - 関係のテンプレートへの記入により, 希望する検索結果の例を示す非手続き問い合わせを作成する。
 - オブジェクト指向問い合わせを C++ や Java のような独立型言語に埋め込む (例えば, SELECT Col.Method() FROM Object) 。

IM6 関係データベース設計 [選択]

トピックス:

データベース設計

関数従属性

正規形 (第 1 正規形 (1NF), 第 2 正規形 (2NF), 第 3 正規形 (3NF), ボイスコッド正規形 (BCNF))

多値従属性 (第 4 正規形 (4NF))

結合従属性 (射影結合正規形 (PJNF), 第 5 正規形 (5NF))

表現理論

学習の目標:

- 関係スキーマの部分集合である二つ以上の属性間の関数従属性を決定する。
- 第 1 正規形, 第 2 正規形, 第 3 正規形およびボイスコッド正規形の意味を述べる。
- 関係が第 1 正規形, 第 2 正規形, 第 3 正規形またはボイスコッド正規形かどうかを特定する。
- 第 1 正規形関係を第 3 正規形 (または, ボイスコッド正規形) 関係に正規化し, 逆に関係スキーマを非正規化する。
- データベース操作の処理効率への正規化の影響, 特に問い合わせ最適化に対する影響を説明する。
- 多値従属性の意味, およびそれがどのような種類の制約を意味するかを述べる。
- 第 4 正規形がスキーマ設計においてなぜ有益であるかを説明する。

IM7 トランザクション処理 [選択]

トピックス:

トランザクション

障害と回復

同時実行制御

学習の目標:

1. SQL をアプリケーションプログラムに埋め込むことによってトランザクションを作成する。
2. 暗黙コミット概念を説明する。
3. 効率的なトランザクション実行に特有な問題について述べる。
4. いつ、なぜロールバックが必要か、そして、ロギングがどのようにロールバックを保証するかを説明する。
5. 同時実行制御機構において様々な分離レベルの効果を説明する。
6. 特定のトランザクションプロトコルの実現に対して、適切な分離レベルを選択する。

IM8 分散データベース [選択]

トピックス:

分散データ格納
分散問合わせ処理
分散トランザクションモデル
同時実行制御
同種解および異種解
クライアントサーバ

学習の目標:

1. 分散データベース設計プロセスにおいて、データ断片化、複製、および割当てに利用される技術を説明する。
2. 分散問合わせを実行するための簡単な手法を評価し、データ転送量を最小限にするものを選ぶ。
3. 複数のノードに格納されたデータベースをアクセスするトランザクションの完了作業において、2相コミットプロトコルがどのように使用されるかを説明する。
4. 特徴的な複製技術や投票方法に基づく同時実行制御を述べる。
5. クライアントサーバモデルにおけるソフトウェアの三つのレベルについて述べる。

IM9 データベースの物理設計 [選択]

トピックス:

記憶域およびファイル構造
索引ファイル
ハッシュファイル
シグネチャファイル
B 木
稠密索引ファイル
可変長レコードのファイル
データベースの効率とチューニング

学習の目標:

1. レコード, レコード型およびファイルの概念, ならびにファイルレコードをディスク上に配置するための技術を説明する。
2. 1次, 2次索引およびクラスタ化索引の適用例を示す。
3. 非稠密索引と稠密索引の区別を明らかにする。
4. B木を利用した動的多段階索引を実現する。
5. 内部および外部ハッシュ技術の理論と応用を説明する。
6. ハッシュを使用して動的ファイル拡張を可能にする。
7. ハッシュ, 圧縮, 効率的なデータベース探索の3者間の関係を示す。
8. 様々なハッシュスキームの費用と効果を評価する。
9. 物理データベース設計がデータベーストランザクションの効率にどれほど影響を及ぼすかを説明する。

IM10 データマイニング [選択]

トピックス:

データマイニングの有用性
 同時パターンおよび順次パターン
 データクラスタリング
 マーケットバスケット分析
 データクリーニング
 データ可視化

学習の目標:

1. 研究・応用で明らかにされてきたデータマイニングに関する様々な概念を比較し, 対照する。
2. 商業的マーケットバスケットデータにおいて, 同時関係を発見することの役割を説明する。
3. 同時関係規則の発見によって得られる様々なパターンの特性を示す。
4. 同時関係規則を得るため, 関係システムをどのように拡張すべきかを述べる。
5. データマイニングの効果的応用に共通した方法論的問題を評価する。
6. 提示されたデータに内在する, 雑音, 冗長性, 例外値の根源を特定し特徴づける。
7. データマイニングプロセスにおいて繰返し防止機構 (オンライン集約, ストリームに対する振る舞い, 対話型可視化) を特定する。
8. 繰返し防止機構によってデータマイニングの有効性が改善される理由を示す。

IM11 情報格納と検索 [選択]

トピックス:

文字, 文字列, 符号化, テキスト
 文書, 電子出版, マーク付け, マーク付け言語
 トライ, 転置ファイル, PAT木, シグネチャファイル, 索引付け

形態素解析, 語幹抽出, 句, 停止語リスト
 用語度数分布, 不確実性, 曖昧さ, 重み付け
 ベクトル空間, 確率モデル, 論理モデル, 高度なモデル
 情報ニーズ, 適合性, 評価, 有効性
 類義語 (シソーラス), オントロジ, 分類と類別化, メタデータ
 書誌情報, 書誌メトリックス, 引用
 ルーティングおよび (コミュニティ) フィルタリング
 探索および探索戦略, 情報追跡行動, ユーザモデリング, フィードバック
 情報要約と可視化
 引用, キーワード, 分類手法, およびその他の用語の統合
 プロトコルおよびシステム (Z39.50, OPAC, WWWエンジン, 研究システムを含む)

学習の目標:

1. 情報格納および検索の基本概念を説明する。
2. 効率のよい情報検索に特有な問題は何であるかを述べる。
3. 各種探索方法の応用例を示し, その適用法が当該応用に適当である理由を説明する。
4. インターネットに基づく調査を行う。
5. 小規模ないし中規模の情報格納および検索システムの設計と実現を行う。

IM12 ハイパーテキストとハイパーメディア [選択]

トピックス:

ハイパーテキストモデル (初期の歴史, ウェブ, Dexter, Amsterdam, HyTime)
 リンク機能, エンジン, (分散) ハイパーテキストアーキテクチャ
 ノード, 複合, アンカー
 次元, 単位, 場所, 範囲
 ブラウジング, ナビゲーション, ビュー, ズーミング
 自動リンク生成
 提示, 変形, 同期
 オーサリング, 読込み, アノテーション
 プロトコルおよびシステム (ウェブ, HTTPを含む)

学習の目標:

1. 初期のバージョンから現在の製品に至る, ハイパーテキストおよびハイパーメディアモデルの発展を要約し, それぞれの能力および限界を明らかにする。
2. ハイパーテキストおよびハイパーメディアの基本概念を説明する。
3. 情報提示, 変形, および同期の基本概念を説明する。
4. 使用されるプロトコルおよびシステムに基づいて, ハイパーメディア配信を比較し, 対照する。
5. 適切なオーサリングツールを使って, ウェブ化が可能な情報検索応用を設計し, 実現する。

IM13 マルチメディア情報とシステム [選択]

トピックス:

装置, デバイスドライバ, 制御信号とプロトコル, DSP (デジタル信号処理)
 応用, メディアエディタ, オーサリングシステム, オーサリング
 ストリーム/構造, 獲得/表現/変形, 空間/領域, 圧縮/符号化
 内容分析, 索引付け, および音声・画像・ビデオの検索
 提示, レンダリング, 同期化, マルチモーダル統合/インタフェース
 リアルタイム配信, サービス品質 (QoS), 音声/ビデオ会議, ビデオ・オン・デマンド

学習の目標:

1. マルチメディア情報およびシステムと通常関連しているメディアおよび支援装置について述べる。
2. 基本的なマルチメディア提示概念を説明する。
3. マルチメディア情報システムにおけるコンテンツ指向内容分析の利用を示す。
4. 音声, ビデオ, 画像, 色, およびその他の情報提示概念の適切な使用の見地から, マルチメディア提示方法を批評する。
5. 市販オーサリングシステムを利用してマルチメディア応用を実現する。

IM14 電子図書館 [選択]

トピックス:

デジタル化, 格納, 交換
 デジタルオブジェクト, 複合物, パッケージ
 メタデータ, カタログ化, 著者送信
 名前付け, 保管庫, アーカイブ
 空間 (概念的, 地理的, 2次元/3次元, VR (仮想現実))
 アーキテクチャ (エージェント, バス, ラッパー/メディアエータ), 相互運用性
 サービス (検索, 連結, ブラウジングなど)
 知的財産権の管理, 機密性, 保護 (透かし挿入)
 集積化と保持, 整合性

学習の目標:

1. 電子図書館の構築の基礎となる技術的概念を説明する。
2. 検索, 連結, およびブラウジングのために必要な基本的サービス要件を述べる。
3. 電子図書館の適切および不適切な利用によって引き起こされるシナリオを批判し, 各シナリオの社会的, 法的および経済的な結果を予測する。
4. 電子図書館の情報の集積化および保持に関連した問題に対する技術的な解答を示す。
5. 小規模な電子図書館の設計と実現を行う。

社会的な観点と職業専門人としての問題 (SP)

- SP1 . コンピュータの歴史 [コア]
- SP2 . コンピュータの社会的文脈 [コア]
- SP3 . 分析の方法およびツール [コア]
- SP4 . 専門上のまた倫理的な責任 [コア]
- SP5 . コンピュータ・ベース・システムのリスクと脆弱性 [コア]
- SP6 . 知的財産権 [コア]
- SP7 . プライバシーと市民的自由 [コア]
- SP8 . コンピュータ犯罪 [選択]
- SP9 . コンピュータにおける経済問題 [選択]
- SP10 . 哲学的枠組み [選択]

あらゆるコンピューティングカリキュラムにおいて、技術的問題が中心となることは明らかであるが、それだけではその分野の完全な教育プログラムを構成することはできない。学生は、コンピュータが現実に機能している社会的・職業的状况についての理解も深めなければならない。

社会的観点からなされた研究成果をカリキュラムに組み込むことの必要性はコンピューティングカリキュラム 1991 [Tucker91] においても認識されていた。この報告の中には、次のように述べられている。

学部学生は、コンピュータ科学に固有の文化、社会、法、倫理上の問題点について、理解する必要がある。この分野がこれまでどのような状況にあったか、現在はどうか、そして将来どこに進んでいくのかを理解しなければならない。また、この進展の中で、彼ら自身の果たすべき役割を自覚するとともに、この学問の発展に重要な役割を果たす哲学上の論点、技術的問題、および審美的価値を十分に理解しなければならない。

学生はさらに、コンピュータの社会的影響について真摯な問いを発し、それに対する答えを評価する能力を磨く必要がある。実務家を志す者は、ある製品をある環境に導入することによってもたらされる影響を予測できなければならない。その製品は、生活の質を高めるのか、低めるのか？ 個人、グループ、組織にどのような影響を与えるだろうか？

最後に学生は、ソフトウェアやハードウェアの販売者と利用者がもつ、法律上の基本的権利に注意する必要がある。また、これらの権利の基となる倫理的価値を認識しなければならない。実務家を志す者は、彼らが負うことになる責任と、失敗した場合に起こりうる結果を理解しなければならない。また、自分自身の限界とツールの限界とを知らなければならない。すべての実務家は、自ら選んだ専門分野とコンピュータ科学分野全体について、その専門能力を長期にわたり最新に保つよう努力し続けなければならない。

この知識分野における教育は、一つの必修科目と、他のいくつかの科目内に含まれた短い教材

を組み合わせることによって最も効果的に行うことができる。コアとして列挙されているユニット (特に SP2, SP3, SP4, SP6) は、他の伝統的な科目では簡単にはカバーしえない。独立した科目を設けずに、これらの項目を適切にカバーすることは難しい。一方で、倫理的な考察が「文脈において」ではなく単独の科目でのみ取り扱われるなら、技術的プロセスには倫理問題は無縁であるという間違った考えを強めてしまうことになるだろう。したがって、伝統的な科目の技術的な題材の文脈の中に、倫理的な問題を分析する部分を取り入れることは重要である。ソフトウェア工学、データベース、コンピュータ・ネットワーク、コンピュータ入門などの分野における科目では、倫理問題を分析するのに適した場が明らかに存在する。しかしながら、倫理に関連する部分は、カリキュラムのほとんどすべての科目に対しても開発されるであろう。独立した科目のみを作ることは、本提言の主旨に明らかに反すると思われる。これらの問題に倫理的また技術的な手段を駆使して前向きに取り組むよう、実務家の責任に訴えるためには、この分野が抱えるあらゆる問題に目を通す必要がある。

授業において論じられる倫理の問題は、その授業の主題に直接関連し、そこから自然に生じるものにすべきである。データベース科目におけるデータ集約やデータマイニングの議論、あるいはソフトウェア工学科目における顧客に対する義務とユーザやその作業によって影響を受ける者に対する義務との間の潜在的な矛盾についての議論、などがその例である。例えば、眼科手術に際してレーザーの動きを制御するアプリケーションを中心に作成されたプログラミング課題は、コンピュータの専門的、倫理的、社会的な影響を考える助けとなるだろう。

コア科目を下級生 (1, 2年生) レベルで行うか、上級生 (3, 4年生) レベルで行うかについては、教育学上の論争があり、未解決である。下級生レベルでこの科目を行うことの利点は次のとおりである。

1. 種々の技術分野における倫理問題を分析するに先立って、分析の方法およびツール (SP3) を履修することを可能とする。
2. 就職のために中途退学する学生でも、専門家としての観点と倫理上の問題のいくつかをそれまでに習得することができる。

一方、この科目をあまりにも早い学年に設けることは、次のような問題を招くかもしれない。

1. 1, 2年レベルの学生は、つきつめた倫理的分析を行えるような技術的知識や知的成熟さに欠けるかもしれない。技術的な選択の可能性を基本的に理解していないと、それらの倫理的意味を考察することは難しい。
2. 倫理上の問題の背景や関連する事項を理解するには、ある程度の成熟さと複雑な思考力が必要である。このため、学生は、少なくとも離散数学科目と、もう一つのコンピュータ科学科目を習得しておくべきである。また、学生がテクニカルライティング科目を取るならば、それは SP 領域の必修科目の前提科目か同時習得科目として指定される必要があるだろう。
3. 教育プログラムによっては、この科目を4年生の「卒業」実践科目として使うことが望ましいと判断するかもしれない。

以下で SP2 と SP3 が必要時間数とともに示されているが、これらは他のすべての項目の基礎となるものである。したがって、教員は他の領域を取り扱う際にも、常に社会的背景にある諸問題と倫理的分析技術に注意を払うべきである。このことは、他の領域における教科が習得されるに従い、SP2 と SP3 の項目が継続的に強化されていくことを意味する。

SP1 コンピュータの歴史 [コア]

最低履修時間:1 時間

トピックス:

先史 (1946年以前の世界)
コンピュータ・ハードウェア, ソフトウェア, ネットワーク化の歴史
コンピュータ科学のパイオニア

学習の目標:

1. コンピュータ科学分野におけるパイオニア達の貢献を列挙する。
2. パーソナルコンピュータとインターネットの出現前と後の日常生活を比較する。
3. コンピュータ分野の歴史における重要な継続的傾向を明らかにする。

SP2 コンピュータの社会的文脈 [コア]

最低履修時間:3 時間

トピックス:

コンピュータの社会的意味
ネットワーク・コミュニケーションの社会的意味
インターネットの成長と制御, インターネットへのアクセス
ジェンダー (性差) に関連する問題
国際問題

学習の目標:

1. 情報処理の一つの実装例についてその社会的文脈を解釈する。
2. 特定の設計に組み込まれた仮定と価値観を特定する。
3. 実験的データを用いて一つの実装例を評価する。
4. コンピュータが人と人の間の対話モードを促進する場合と阻害する場合を記述する。
5. コンピュータやネットワークへのアクセスが制限される国がある理由を説明する。

SP3 分析の方法およびツール [コア]

最低履修時間:2 時間

トピックス:

倫理的な論証の実践と評価
 倫理的な選択肢の同定と評価
 設計の社会的文脈の理解
 仮定と価値観の同定

学習の目標:

1. 論証を分析して、前提と結論を定める。
2. 倫理上の論証において例、類推、および反類推が使用されることを示す。
3. 一つの論証において基本的な論理上の誤りの例を見つけ出す。
4. 一つの問題における利害関係者とその人たちに対するわれわれの義務を定める。
5. 技術的決定における倫理上のトレードオフを明確にする。

SP4 専門上のまた倫理的な責任 [コア]**最低履修時間:3 時間****トピックス:**

われわれが生活するコミュニティにおける価値観と法
 専門家とは何か
 様々な形式による専門家の資格認定、その利点と欠点
 公共政策における専門家の役割
 結果に対しての持続的注意
 倫理上の反対意見と内部告発
 倫理綱領、行動、実践 (IEEE, ACM, SE, AITPなど)
 ハラスメントと差別への対応
 職場のコンピュータ環境における「許容される利用」規定

学習の目標:

1. 内部告発事件の進展段階を識別する。
2. プロ意識の表現および意思決定の手引として、関連する専門家綱領の長所と短所を明示する。
3. ソフトウェア開発において起こる倫理上の問題を識別し、それに対する技術的、倫理的な対策を決定する。
4. 強制を伴うコンピュータ使用方針を策定する。
5. 問題进行处理の際の専門家と政府官僚の役割に着目して、グローバルなコンピュータ問題を分析する。
6. ACM, IEEE コンピュータソサエティ、その他の組織の専門家倫理綱領を評価する。

SP5 コンピュータ・ベース・システムのリスクと脆弱性 [コア]

最低履修時間:2 時間

トピックス:

ソフトウェア・リスクの歴史的な例 (Therac-25のケースなど)
ソフトウェアの複雑さの意味すること
リスクの評価と管理

学習の目標:

1. 正当性を保証する方法としてのテストの限界を説明する。
2. 正当性, 信頼性, 安全性の違いを述べる。
3. 既存のコンポーネントを再利用する際の隠された問題の可能性について論じる。
4. 現在のリスク管理の手法をいくつか示し, その長所と短所を特徴づける。

SP6 知的財産権 [コア]

最低履修時間:3 時間

トピックス:

知的財産権の基礎
著作権, 特許権, 企業秘密
ソフトウェア著作権侵害
ソフトウェア特許
知的財産権に関する国際間の問題

学習の目標:

1. 特許権, 著作権, および企業秘密保護の差異を明らかにする。
2. 国内法および国際法における著作権の法的背景について論じる。
3. 特許権法および著作権法が国によってどのように異なっているかを説明する。
4. ソフトウェア特許の歴史的な進展を概説する。
5. ソフトウェア著作権侵害のソフトウェア開発者への影響, および関連取り締まり機関の役割について論じる。

SP7 プライバシーと市民的自由 [コア]

最低履修時間:2 時間

トピックス:

プライバシー保護のための倫理的法的根拠
大規模データベースシステムにおけるプライバシーの意味

プライバシー保護のための技術戦略
サイバースペースにおける表現の自由
国家間や異文化間にもたらす影響

学習の目標:

1. 自国におけるプライバシーの権利と表現の自由のための法的根拠, および, これらの概念が国によってどのように異なるかについて要約する。
2. 現在のコンピュータによるプライバシーへの脅威について述べる。
3. インターネットが表現の自由保護に関する歴史的なバランスをいかに変化させうるかを説明する。
4. サイバースペースにおける表現の自由の不利益と利益の両方について説明する。
5. 技術において示されたプライバシー保護の傾向を述べる。

SP8 コンピュータ犯罪 [選択]

トピックス:

コンピュータ犯罪の歴史および事例
「クラッキング」(ハッキング) とその効果
ウイルス, ワーム, トロイの木馬
犯罪防止戦略

学習の目標:

1. ウイルスやサービス不能化攻撃の技術的な原理を概説する。
2. 「クラッカー」攻撃と戦うための技術を列挙する。
3. いくつかの異なる「クラッキング」の方法と動機について論じる。
4. セキュリティとそれに関連するトレードオフに対する専門家の役割を識別する。

SP9 コンピュータにおける経済問題 [選択]

トピックス:

独占とその経済的意味
コンピュータ製品の品質に対する熟練労働者の需要と供給の影響
コンピュータ分野における価格決定戦略
コンピュータ資源へのアクセスの差異とそれによる効果

学習の目標:

1. 独占とその経済的な意味を要約する。
2. 労働力供給の不足が情報技術産業にもたらす影響の仕方をいくつか挙げ論じる。
3. コンピュータ・アクセスに対する制約への対処法を提案し, その効果を論ずる。

4. コンピュータ製品とサービスの価格決定戦略の発展を概説する。

SP10 哲学的枠組み [選択]

トピックス:

哲学的枠組み, 特に功利主義と義務論的学説
倫理的相対主義の問題
歴史的な展望における科学倫理
科学的アプローチと哲学的アプローチの差異

学習の目標:

1. 相対主義, 功利主義, 義務論的学説の基本概念を要約する。
 2. 倫理学と職業倫理の違いを認識する。
 3. 倫理的枠組みとしての「雇われエージェント」アプローチ, 厳密な法律尊重主義, 素朴利己主義, および素朴相対主義の弱点を明らかにする。
-

ソフトウェア工学 (SE)

- SE1 . ソフトウェア設計 [コア]
- SE2 . API の使用 [コア]
- SE3 . ソフトウェアツールおよび環境 [コア]
- SE4 . ソフトウェアプロセス [コア]
- SE5 . ソフトウェア要求および仕様 [コア]
- SE6 . ソフトウェア妥当性検査 [コア]
- SE7 . ソフトウェアの進化 [コア]
- SE8 . ソフトウェアプロジェクト管理 [コア]
- SE9 . コンポーネントベース開発 [選択]
- SE10 . 形式手法 [選択]
- SE11 . ソフトウェアの信頼性 [選択]
- SE12 . 専用システムの開発 [選択]

ソフトウェア工学は、ユーザおよび顧客の要求を満足させるように、ソフトウェアシステムを効果的かつ能率的に構築するための理論、知識および実践の具体的な適用に関する専門分野である。ソフトウェア工学は、小規模、中規模、大規模のいずれのシステムにも適用される。また、ソフトウェアシステムのライフサイクルのすべてのフェーズにわたる。このライフサイクルは、要求分析と仕様、設計、構築、テスト、および運用と保守からなる。

ソフトウェア工学は、工学的な手法、処理、技術、および計測法を用いる。また、様々なツールを有効に活用する。すなわち、ソフトウェア開発管理、ソフトウェア製品の分析とモデリング、品質の評価と制御のためのツール、そしてソフトウェアの進化と再利用のための規律正しく制御されたアプローチを保証するツールである。個人の開発者による場合でも、開発チームで行われる場合でも、ソフトウェア開発には、与えられた開発環境のために最も適切なツール、方法、およびアプローチを選択することが要求される。

ソフトウェア工学の各要素は、ソフトウェアシステムを構築するのに、プロ意識、品質、スケジュール、およびコストが重要であるようなどんなコンピュータの適用領域のソフトウェア開発に対しても、適用可能である。

SE1 ソフトウェア設計 [コア]

最低履修時間:8 時間

トピックス:

- 基本的な設計概念および原理
- 設計パターン
- ソフトウェアアーキテクチャ
- 構造化設計
- オブジェクト指向分析および設計

コンポーネントレベル設計 再利用のための設計

学習の目標:

1. よいソフトウェア設計のもつべき性質について論じる。
2. オブジェクト指向分析・設計と構造化分析・設計とを比較対照する。
3. 主要な設計原理および概念に基づいて複数のソフトウェア設計の品質を評価する。
4. ソフトウェアアプリケーションの構築において、適切な設計パターンを選択し、適用する。
5. ソフトウェア要求仕様、一般に認められたプログラム設計方式（例えば、構造化設計やオブジェクト指向設計）および適切な設計表記法を用いて中規模ソフトウェア製品のソフトウェア設計を行い、それを記述する。
6. 適切なガイドラインを利用して、ソフトウェア設計のレビューを行う。
7. コンポーネントレベルでソフトウェアの設計を評価する。
8. 再利用の観点からソフトウェアの設計を評価する。

SE2 API の使用 [コア]

最低履修時間:5 時間

トピックス:

API プログラミング
クラスブラウザおよび関連のツール
例によるプログラミング
API 環境におけるデバッグ
コンポーネントベース開発入門

学習の目標:

1. ソフトウェア開発における API の価値を説明する。
2. API を使うアプリケーションの開発にクラスブラウザおよび関連ツールを使用する。
3. 大規模 API パッケージを使うプログラムの設計、開発、テスト、デバッグを行う。

SE3 ソフトウェアツールおよび環境 [コア]

最低履修時間:3 時間

トピックス:

プログラミング環境
要求分析および設計モデリングツール
テストツール
構成管理ツール

ツール統合メカニズム

学習の目標:

1. ある範囲のソフトウェア製品の開発を支援するために適切なツールの組合せを, その理由を示して選択する。
2. 与えられた領域におけるソフトウェア開発に対し, ツールの組合せ (例えば, マネジメント, モデリング, またはテスト用) を分析して, 評価する。
3. 中規模のソフトウェア製品の開発を支援するようなソフトウェアツールを使う能力のあることを, 明らかにする。

SE4 ソフトウェアプロセス [コア]

最低履修時間:2 時間

トピックス:

ソフトウェアライフサイクルおよびプロセスモデル
プロセス評価モデル
ソフトウェアプロセス計量尺度

学習の目標:

1. ソフトウェアライフサイクルとその各フェーズを, フェーズごとに作成される成果物とともに説明する。
2. 広い範囲のソフトウェア製品の開発および保守に最も適したソフトウェア開発モデルを, その理由を示して選択する。
3. プロセス成熟度モデルの役割を説明する。
4. 伝統的なウォーターフォールモデルを, インクリメンタルモデル, オブジェクト指向モデルを初めとする他の適当なモデルと比較する。
5. 様々なソフトウェアプロジェクト・シナリオの各々について, ソフトウェアライフサイクルにおけるプロジェクトの位置付けを述べ, 次に遂行されるべき特定のタスクを定め, またそれらのタスクに適した計量尺度を定める。

SE5 ソフトウェア要求および仕様 [コア]

最低履修時間:4 時間

トピックス:

要求獲得
要求分析モデル化技法
機能よび非機能要求
プロトタイピング

形式仕様技法の基礎的な概念

学習の目標:

1. 要求獲得および分析の主要要素と共通手法を適用し、中規模のソフトウェアシステムのためのソフトウェア要求仕様書を作成する。
2. 遺産ソフトウェアを保守することの難しさについて論じる。
3. 普通の非形式手法を使用して、中規模ソフトウェアシステムのための要求を要求仕様書の形でモデル化し、記述する。
4. 文書の品質を決定するための最良の方法によって、ソフトウェア要求書のレビューを行う。
5. 広く使用されている形式仕様言語で書かれたソフトウェア要求仕様書を自然言語に変換する。

SE6 ソフトウェア妥当性検査 [コア]

最低履修時間:3 時間

トピックス:

妥当性検査 (validation) 計画
テスト基本事項 (テスト計画立案およびテストケース生成を含む)
ブラックボックスおよびホワイトボックステスト技術
単体テスト, 統合テスト, 妥当性確認テスト, システムテスト
オブジェクト指向テスト
査閲 (inspection)

学習の目標:

1. プログラムの妥当性検査と検証とを区別する。
2. ソフトウェアの妥当性検査においてツールの果たしうる役割を示す。
3. 中規模のソフトウェア製品に対する様々なタイプおよびレベルのテスト (単体, 統合, システム, 受入れの各テスト) の違いを示す。
4. 中規模コードセグメントのテスト計画を作成し, 評価し, 実行する。
5. チーム活動の一部として中規模コードセグメントの査閲を行う。
6. オブジェクト指向ソフトウェアのテストが抱える問題点について議論する。

SE7 ソフトウェアの進化 [コア]

最低履修時間:3 時間

トピックス:

ソフトウェアの保守
保全性のあるソフトウェアの特徴

リエンジニアリング
遺産システム
ソフトウェアの再利用

学習の目標:

1. ソフトウェアの進化と関連している主要な問題を確認し、それらのソフトウェアライフサイクルに対する影響を説明する。
2. 遺産システムを保守することにかかる問題点、およびリバースエンジニアリングの必要性を論じる。
3. 回帰テストのプロセスおよびリリース管理におけるその役割を概説する。
4. 中規模のある既存の製品に対する変更要求の影響を見積る。
5. ある変更要求に応じて中規模の製品をリエンジニアリングするための計画を立案する。
6. ソフトウェア再利用の利点および欠点について論じる。
7. 与えられた文脈においてソフトウェア再利用の機会を活用する。

SE8 ソフトウェアプロジェクト管理 [コア]

最低履修時間:3 時間

トピックス:

チーム管理

- チームプロセス
- チーム編成と意思決定
- ソフトウェアチームにおける役割と責任
- 役割の識別と割当て
- プロジェクト追跡
- チーム問題解決

プロジェクト計画

ソフトウェアの計測および見積り技術

リスクの分析

ソフトウェア品質保証

ソフトウェア構成管理

プロジェクト管理ツール

学習の目標:

1. チームプロジェクトへの参加を通して、チーム構成とチーム管理の主要な要素を明らかにする。
2. ソフトウェアプロジェクトに備えてプロジェクト計画を準備する。この計画には規模および工数の見積り、日程、資源配分、構成制御、変更管理、およびプロジェクトリスクの認識と管理が含まれる。

3. ソフトウェア製品の品質を保証するために使用される様々な方法と技術を比較対照する。

SE9 コンポーネントベース開発 [選択]

トピックス:

基本的事項

- コンポーネントの定義および性質
- コンポーネントとインターフェース
- 契約としてのインターフェース
- コンポーネントの利点

基礎技術

- コンポーネントの設計および組立て
- クライアントサーバモデルとの関係, パターンとの関係
- オブジェクトおよびオブジェクトライフサイクル・サービスの使用
- オブジェクトブローカの使用
- マーシャリング

応用 (モバイルコンポーネントの使用を含む)

コンポーネントベースシステムのアーキテクチャ

コンポーネント指向設計

イベント処理: 検出, 通知, 応答

ミドルウェア

- ミドルウェア中のオブジェクト指向パラダイム
- オブジェクトリクエスト・ブローカ
- トランザクション処理モニタ
- ワークフローシステム
- 最先端のツール

学習の目標:

1. 高品質のソフトウェアコンポーネントの構築における既存の原理を説明し, 適用する。
2. 与えられたシナリオに適切なコンポーネントベースシステムのためのアーキテクチャについて論じ, 選択する。
3. 与えられた一つ以上の API において実装されているイベント処理の種類を識別する。
4. ミドルウェアシステムにおけるオブジェクトの役割およびコンポーネントとの関係を説明する。
5. コンポーネント指向アプローチを種々のソフトウェアの設計に適用する。ソフトウェアとしては並行処理とトランザクション, 信頼できる通信サービス, 遠隔照会およびデータベース管理のためのサービスを含むデータベース操作, 安全な通信とアクセスなどのためのものを含む。

SE10 形式手法 [選択]

トピックス:

形式手法の概念
形式仕様言語
実行可能な仕様および実行不可能な仕様
事前表明および事後表明
形式的検証

学習の目標:

1. 簡単なソフトウェアセグメントに形式的検証技術を適用する。
2. ソフトウェアの妥当性検査およびテストにおける形式的検証技術の役割について論じる。
3. 形式仕様言語を用いることの潜在的な利益および不利益を説明する。
4. 簡単なものから複雑なものにわたる様々な状況に対して事前および事後表明を作成し、評価する。
5. 一般に使われている形式仕様言語を用いて、簡単なソフトウェアシステムの仕様書を定式化し、品質の見地からその利点を示す。

SE11 ソフトウェアの信頼性 [選択]**トピックス:**

ソフトウェアの信頼性モデル
冗長性とフォールトトレランス
欠陥の分類
確率的分析方法

学習の目標:

1. あるソフトウェアシステムの信頼性を見積りを行うために、複数の方法を適用する能力を示す。
2. 中規模のアプリケーションに必要とされる冗長性とフォールトトレランスを定め、実際にその機能をもたせる。
3. 非常に高いレベルの信頼性を達成するに当たって存在する問題点を説明する。
4. 指定された信頼性のレベルを達成するソフトウェアアーキテクチャを実現することができる方法を同定する。

SE12 専用システムの開発 [選択]**トピックス:**

リアルタイムシステム
クライアントサーバシステム
分散処理システム

並列システム
ウェブベースシステム
高完成度のシステム

学習の目標:

1. 様々な専用システムを識別し, 論じる。
 2. 特定の目的のために設計されているソフトウェアシステムに関して, ライフサイクルおよびソフトウェアプロセスの問題点について論じる。
 3. 専用のソフトウェアシステムの効率的かつ効果的な開発および保守をもたらすと思われる方法を選び, その理由を示す。
 4. 特定の状況で関連する専門的な一群の問題が与えられたとして, 専用システムの開発に従事するソフトウェア技術者がそれらの問題にいかに対応すべきであるかを議論する。
 5. 専用システム開発の実施と関連した中心的な技術上の問題を概観する。
-

計算科学と数値計算 (CN)

- CN1 . 数値解析 [選択]
- CN2 . オペレーションズリサーチ [選択]
- CN3 . モデリングとシミュレーション [選択]
- CN4 . ハイパフォーマンス・コンピューティング [選択]

数値計算法と科学計算技術は、最も早い時期から、コンピュータ科学の研究の主要な分野を成してきた。コンピュータが問題解決能力を増すに従い、この分野は、コンピュータ科学のほとんどの分野と同じく、広がりと重要性の両面において成長した。20世紀末において、科学計算は、コンピュータ科学と密接に関係しながらもそれと一線を画する、それ自身において知的な専門分野として成立している。

数値計算法と科学計算の諸科目は、コンピュータ科学における学部学生の教育課程の非常に価値ある構成要素ではあるが、CC2001 タスクフォースは、この分野におけるこれらの科目のいずれもコア知識とはならないと考えている。カリキュラムの調査およびコンピュータ科学教育コミュニティとのやりとりから、すべてのコンピュータ科学を学ぶ学生にとってこの教科が必要不可欠であるというコンセンサスは存在しないと、われわれは確信している。この教科はコンピュータ科学の重要な一部ではあるが、すべての教育プログラムにとっての一部である必要はない。

この教科を学習することを選ぶ人にとっては、この分野は多くの価値あるアイデアと技術を学ぶ機会を提供する。その中には、例えば、数値表現の精度、誤差解析、数値技術、並列アーキテクチャと並列アルゴリズム、モデリングとシミュレーション、および科学的可視化が含まれる。それと同時に、この分野のコースを選択する学生はこれらの技術を次のような広い範囲の応用領域に適用する機会をもつことができる。

- 分子力学
- 流体力学
- 天体力学
- 経済予測
- 最適化問題
- 材料の構造解析
- 生物情報科学
- 計算生物学
- 地質学上のモデリング
- コンピュータ断層撮影

この領域 (CN) における各ユニットは、ほとんどの教育機関において 1 学期コースに相当する。したがって、他の領域で使用されているような、個々のユニットがもっと少ない時間を必要とするケースとは、トピックスの記述および学習目標の設定のレベルが異なっている。

CN1 数値解析 [選択]

トピックス:

浮動小数点演算
 誤差, 安定性, 収束
 テイラー級数
 求解のための逐次的解法 (ニュートン法)
 曲線の当てはめ: 関数近似
 数値微分と数値積分 (シンプソン則)
 陽および陰な計算法
 微分方程式 (オイラー法)
 線形代数
 差分法

学習の目標:

1. このユニットにおいて提示される数値解析法を比較対照する。
2. 誤差, 安定性, 機械精度の概念, および計算近似の非正確性の概念について定義を述べる。
3. 計算近似の非正確性の生じる原因を見つけ出す。
4. 数値計算法を実行するプログラムを設計し, コーディング, テスト, デバッグする。

CN2 オペレーションズリサーチ [選択]**トピックス:**

線形計画法

- 整数計画法
- シンプレックス法

 確率的モデリング
 待ち行列理論

- ペトリネット
- マルコフモデルと連鎖

 最適化
 ネットワーク解析とルーティングアルゴリズム
 予測と推定

- 決定解析
- 予測
- リスク管理
- 計量経済学, ミクロ経済学
- 感度分析

 動的計画法
 応用例
 ソフトウェアツール

学習の目標:

1. オペレーションズリサーチの基本的な技法を適用する。
2. 予測および推定のためのいくつかの確立されている技法を述べる。
3. オペレーションズリサーチの分野における問題を解決するために応用プログラムを設計し、コーディング、テスト、デバッグする。

CN3 モデリングとシミュレーション [選択]

トピックス:

乱数

- 擬似乱数生成とテスト
- モンテカルロ法
- 分布関数初歩

シミュレーションモデリング

- 離散イベントシミュレーション
- 連続シミュレーション

シミュレーションモデルの検証と妥当性実証

- 入力分析
- 出力分析

待ち行列モデル

応用例

学習の目標:

1. 計算機シミュレーションの基本的な概念について論じる。
2. 計算機シミュレーションのモデルを評価する。
3. 乱数生成法を比較対照する。
4. シミュレーションプログラムを設計し、コーディング、テスト、デバッグする。

CN4 ハイパフォーマンス・コンピューティング [選択]

トピックス:

ハイパフォーマンス・コンピューティング初歩

- 計算科学の歴史と重要性
- 応用分野の概観
- 必要な技法のレビュー

ハイパフォーマンス・コンピューティング

- プロセッサ・アーキテクチャ
- 高性能化のためのメモリシステム
- 入出力装置
- パイプライン処理

- 並列言語と並列アーキテクチャ

科学的可視化

- 結果のプレゼンテーション
- データ形式
- 可視化ツールとパッケージ

問題例

- 大洋および大気モデル
- 地震波伝播
- N体システム (バーンズ・ハットアルゴリズム)
- 化学反応
- 相転移
- 流体の流れ

学習の目標:

1. 計算モデリングが現在の研究手法を向上させるような問題分野を識別する。
 2. 科学並列計算の各種アーキテクチャについて、それぞれの長所と短所を認識しつつ、比較対照する。
 3. ハイパフォーマンス・システムのための簡単な性能測定を実現する。
 4. 数値解析、計算機シミュレーション、および科学的可視化の技法を使用したプログラムを設計し、コーディング、テスト、デバッグする。
-