

解説



情報処理専門教育について

大学等における情報系専門教育の改善への提言[†]

大学等における情報処理教育検討委員会 CS 分科会

野口 正 一¹⁾ (委員長) 牛島 和 夫²⁾ (CS 主査) 榎 本 彦 衛³⁾
 木 村 泉⁴⁾ 高 橋 延 匡⁵⁾ 都 倉 信 樹⁶⁾ 諸 橋 正 幸⁷⁾
 中 森 眞理雄⁵⁾ (CS 幹事)

1. はじめに

社会の高度情報化が急速に進行しつつある今、高度な知的集約産業の確固たる基盤を構築することは、我が国にとって急務である。その中心的課題は、情報工学・科学の諸分野において世界の指導的地位を占めることであり、それなくしては、我が国の将来はきわめてきびしいものとなろう。しかるに、我が国が目指すべき知的所有権による立国を支える人材はこの分野できわめて不足しており、大学等においても産業界においても基本的な知的産物は大半を海外に頼っているのが実状である。これはきわめて由々しいことであり、情報工学・情報科学の分野の人材の養成・確保が急がれるゆえんである。

近年、我が国では、多くの大学等において、情報系専門学科が数多く設置されている。しかし、

そこでの教育を担当する真の専門家の数はきわめて少なく、またカリキュラムをはじめ多くの重要な問題がほとんど未解決のまま残されているのが実状である。

このような状況のもとで、情報処理学会は文部省の委託を受け、長期的展望のうえに具体的なカリキュラムを検討することを目的とした「情報処理教育の改善のための調査研究」を「大学等における情報処理教育検討委員会」を設けて実施し、暫定モデルカリキュラム案を作成した。

本提言では、2. においてこの調査研究の主な成果を手短かに要約するとともに、3. において我が国における情報系専門教育の問題点に関し、若干の現状分析を行う。

2. 暫定モデルカリキュラム案

情報工学・情報科学は情報処理の基礎と応用に関する系統的な学問であり、問題のモデル化・定式化、問題解決の手順（アルゴリズム）、仕掛の構造（アーキテクチャ）、システムの構築方法などを不可欠の内容とするものであり、その基盤は学問としての計算機分野にある。

当委員会では、まず、我が国と米国の大学の情報系専門学科のカリキュラムを検討した。その結果、我が国においても優れたカリキュラムをもつ大学がいくつか見いだされたものの、全体としては、我が国の大学の情報系カリキュラムは米国などの大学の情報系カリキュラムと比較すると著しく劣ることが判明した。特徴的なことは、前者が計算機や情報処理と無関係なものまで含めた広範囲なものとなっており、学問体系としての存在基

[†] Proposals and Curriculum Recommendations for B.A. Degree Programs in Computer Science and Engineering by CS Working Group of the Curriculum Committee for B.A. Degree in Computer Science and Information Engineering, Committee Chair: Shoichi NOGUCHI (Tohoku University, Research Center of Applied Information Science), CSWG Chair: Kazuo USHIJIMA (Department of Computer Science and Communication Engineering, Kyushu University), Hikoe ENOMOTO (Department of Mathematics, Keio University), Izumi KIMURA (Department of Information Science, Tokyo Institute of Technology), Nobumasa TAKAHASHI (Department of Computer Science, Tokyo University of Agriculture and Technology), Nobuki TOKURA (Department of Information and Computer Sciences, Osaka University), Masayuki MOROHASHI (Tokyo Research Laboratory, IBM Japan Ltd.), and CSWG Vice-chair: Mario NAKAMORI (Department of Computer Science, Tokyo University of Agriculture and Technology).

1) 東北大学応用情報学研究所、2) 九州大学工学部情報工学科、3) 慶應義塾大学理工学部数理科学科、4) 東京工業大学理工学部情報科学科、5) 東京農工大学工学部電子情報工学科コンピュータサイエンスコース、6) 大阪大学基礎工学部情報工学科、7) 日本アイ・ビー・エム(株)東京基礎研究所

盤が薄いということである。このようなカリキュラムは残念ながら国際的通用性に欠け、学生や教員の国際交流の障害となり、我が国が学問分野で世界に貢献する可能性を自ら狭めるものである。

米国計算機学会 (Association for Computing Machinery, 以下 ACM と略す) では、1968 年以来 10 年ごとに大学の計算機分野のモデルカリキュラムに関する提案・提言を行っている。計算機分野が独立した学問であることを明らかにした 68 年提案¹⁾、今日でもかなりの部分が通用する 78 年提案²⁾、学問分野としての枠組を示した 88 年提言³⁾ (およびそれに基づくカリキュラム案の例⁴⁾) がこれまでに発表されている。

当委員会は、急増する情報系専門学科の新設に対応してモデルカリキュラムを緊急に発表する必要から、暫定モデルカリキュラム案 “IPSJ CS カリキュラム J90” (以下、カリキュラム J90 と略す) を作成した。カリキュラム J90 は、骨格は ACM 78 年提案にほぼ対応し、内容は ACM 88 年提言を反映したものである。ただし、骨格・内容ともに、我が国独自の事情や近年の学問の進歩を加味してある。日本語情報処理、情報ネットワーク学がそれである。カリキュラム J90 は情報系専門学科で教えるべき計算機分野の授業科目の最小限のモデルを示したものであり、それ以外の授業科目をも教えることを排除するものではない。また、カリキュラム J90 は教えるべき内容をまとめて授業科目の編成・名称を例示したものであり、固定的なものではない。カリキュラム J90 の内容を活かして、独自の編成のカリキュラム、独自の領域を加味したカリキュラムを作るなど、学科の創意と個性が望まれる。

当委員会としては、外国に依存せず世界にインパクトを与える我が国独自のカリキュラムを作成することを目標として、さらに検討を続ける計画である。

2.1 モデルカリキュラムが想定する学科

前述のとおり、近年我が国の多くの大学で情報系専門学科が設置されている。それらの中には、さまざまな分野と情報工学・情報科学との境界領域に主眼をおいているとみられる学科もあり、個性的な名称の学科も多い。

当委員会ではモデルカリキュラムを作成するにあたり、特定のタイプの学科をモデルに高度に専

門的なカリキュラムを作ることは避け、さまざまな特徴をもつ広範囲の情報系専門学科に共通に使える最小限のカリキュラムを示すにとどめることにした。この考え方は“コアカリキュラム”の 7 科目に現れている (コアカリキュラムという用語は ACM 78 年提案²⁾でも用いられている)。このコアカリキュラムの履修を終えた学生にさらに進んだ計算機分野の授業科目を履修させるための目安として、8 科目を示すことにした。

カリキュラム J90 は教えるべき内容を示したものであり、必修科目数や科目名称などの形式面を規定するものではない。カリキュラム J90 の各授業科目の目的と内容についての以下の記述は J90 のフィロソフィを示すものと解釈されたい。

2.2 暫定モデルカリキュラム案一

カリキュラム J90

カリキュラム J90 は 15 の授業科目からなる。このうち、コアカリキュラム (専門学科で教えるべき最小限のカリキュラム) は次の 7 科目である。情報系専門学科の学生はコアカリキュラムに対応する内容をすべて履修すべきである。

JCS 1 プログラミング序論

JCS 2 プログラムの設計と実現

JCS 3 計算機システム序論

JCS 4 計算機ハードウェア基礎

JCS 5 情報構造とアルゴリズム解析

JCS 6 オペレーティングシステムとアーキテクチャ I

JCS 7 プログラミング言語の構造

次の JCS 8 から JCS 15 までは専門性がやや強く、情報系専門学科の学生はこれらの中から選択的に半分程度を履修し、特に 1 科目を深く履修することが望ましい。しかしながら、情報系専門学科の卒業生であれば、JCS 8 から JCS 15 で取りあげられる内容全般について概括的な知識を有していることが必要である。そのためには情報工学概論 (仮称) のような先修科目を設けて、JCS 8~JCS 15 で教える内容への導入教育を行うような工夫を学科の実状に応じて計画することが望ましい。

JCS 8 オペレーティングシステムとアーキテクチャ II

JCS 9 ファイルとデータベースシステム

JCS 10 人工知能

JCS 11 ヒューマンインタフェース

JCS 12 計算のモデルとアルゴリズム

JCS 13 ソフトウェアの設計と開発

JCS 14 プログラミング言語の理論と実際

JCS 15 数値計算の理論と実際

カリキュラム J 90 の授業科目の間には、あらかじめ履修しておくべきことを指定した先修条件がある。ただし、授業の進め方など運用を工夫すれば、先修条件が指定された授業科目同士を並行して開講することも可能である。カリキュラム J 90 の各授業科目 JCS 1~JCS 15 の目的・内容と先修条件は、付録 1 に示すとおりである。

2.3 カリキュラム評価法

我が国と米国の大学の情報系専門学科のカリキュラムを共通の尺度で評価するために、それらのカリキュラムを ACM 88 提言で採用されている情報工学・情報科学（同提言では計算機分野と呼んでいる）の副領域とパラダイムにあてはめてみる作業は効果的である。

同提言では計算機分野を次の副領域に分割している。

1. アルゴリズムとデータ構造
2. プログラミング言語
3. アーキテクチャ
4. 数値的および記号的計算
5. オペレーティングシステム
6. ソフトウェア方法論とソフトウェア工学
7. データベースと情報検索
8. 人工知能とロボティクス
9. 人とコンピュータのコミュニケーション

また、同提言では計算機分野に、理論、抽象化、設計の三つのパラダイムがともに重要であると指摘している。これらの副領域とパラダイムに基づいて、同提言では計算機分野に対して 9×3 の定義マトリックス（以下、ACM 88 提言をまとめた委員会の長の名にちなんで「デニング図」と呼ぶ）を提唱している（付録 2 参照）。

デニング図は計算機分野の展望を網羅的に示したものである。カリキュラムを評価するのに便利な方法である。すなわち、27 個の枠にそれぞれの内容がどの程度教えられているかを記入することにより、そのカリキュラムの特徴や傾向を知ることができる。デニング図中において、分布面積の大きさはそのカリキュラムの広さを、分布の偏りと集中度はそのカリキュラムの個性を表す。

実際、米国には、この方法でカリキュラムを自己評価している大学がある。デニング図が唯一無二で最良の方法とは言えないが、国際的に共通の尺度でカリキュラムを比較する便利な方法であるので、広く活用されることが望まれる。

各情報系専門学科とも、デニング図を用いてカリキュラムの自己評価を試みることを推奨する。

2.4 実験・演習テーマ例、試験問題例

情報系専門学科のアイデンティティは実験・演習に現れる。

実験・演習を効果的に実施するには基本的環境を整備しておく必要がある。ハードウェア環境や指導スタッフだけでなく、ソフトウェア環境が重要である。たとえば、日本語情報処理で出会う基本的な問題は実験・演習に積極的に取り込むのがよいが、それには形態素解析プログラムや機械可読な国語辞書などが不可欠である。

試験問題も情報系専門学科のアイデンティティを表す指標の一つである。大学の情報系専門学科の卒業生なら当然身につけているべき知識や能力を問う試験問題の例を付録 3 に掲載する。

3. 我が国の情報系専門教育の問題点

3.1 情報工学・情報科学の理解が不十分である

我が国の情報系専門学科の最大の問題点は、専門性についての理解が教員の間に不足していることである。原因の一つは「情報工学・情報科学」という用語にある。

前述のとおり、情報工学・情報科学は情報処理の基礎と応用に関する系統的な学問であり、計算機分野を基盤とするものである。現実には、情報工学・情報科学という言葉に多様な解釈がなされており、時には計算機や情報処理とは無関係のものまで含めて拡大解釈された例が見受けられる。拡大解釈された情報工学・情報科学の教育は、学問としての計算機分野の含まれ方が少なく、系統的な学問という性格が弱く、専門性が薄い。カリキュラムに、情報系専門学科としてのアイデンティティを主張できる計算機分野固有の授業科目が少ない。

3.2 情報系専門教員の不足

ACM 78 提案では、情報工学・情報科学のコアカリキュラムを教えるだけで、少なくとも 6 名の

教員が必要であると指摘している。大学院まで担当するのであればさらに何人もの専門の教員が必要である。しかし、情報工学・情報科学は歴史の浅い学問であるため、1学科でこれだけの人数の教員を揃えることは容易なことではない。

情報系専門教員には計算機システムの環境整備や維持・管理の業務も期待されるのが普通であり、専門教員が少ないと、それらの業務の負荷が教育・研究という本来の職務を上回る。特に、ネットワーク環境のもとでの計算機システムの維持・管理は高度の専門能力を要し、技術職員だけでは対処できないので、情報系専門教員の負担は大きい。

3.3 専門性の薄い大学科

近年、国立大学ではいわゆる大講座制（一つの講座に複数の教授、助教授、助手を置く）を採用した、大学科への改組が盛んに行われている。工学部の場合、情報関連の学科では、他分野の学科と情報系専門学科とを合併して、「××情報工学科」などとなることが多い。その結果、入学定員250名という大きな学科も珍しくない。

しかし、現実には改組にともなう副作用の防止など、難問が多い。最大の副作用は、当事者の利害から、改組に際して情報工学・情報科学とは無関係の分野が「情報系専門学科を強化する」という名目で取り入れられ、その結果、これから充実しなければならない情報系専門学科の教育の専門性と質が低下することである。もともと計算機分野の教育・研究の含まれ方が少なかった我が国の情報系専門学科が改組によりますます専門性を薄めかねない傾向にあるのは重大である。

3.4 ブラックボックス化が進む教育用計算機

情報系専門学科の主要な教育設備である計算機システムは、今日ではあまりにも巨大なシステムになってしまっているため、学生が全貌を捉えることは不可能になっている。また、著作権問題もあり、卒業後に将来計算機システムの開発に従事するかもしれない学生のために基本ソフトウェアの詳細を公開してもらうことは期待し難い。その結果、学生の実験・演習は計算機システムをブラックボックスとして使うものに重点が移りがちで、プログラミング言語の文法やコマンドの使い方など操作方法の教育が主になってしまう。この点が我が国と米国の情報系専門学科の教育の大き

な差である。

大切なのは学問としての計算機分野のフィロソフィを教育することであり、ブラックボックス化は排除しなければならない。それには、大学が独自に教育用基本ソフトウェアを開発するくらいの意欲が必要である。

3.5 費用のかかる情報系専門教育

近年、パソコンやワークステーションが安価になってきているため、情報系専門教育は少ない経費で実施できると誤解される恐れがある。

MIPS 値などの単純な比較でなく、メモリ空間の大きさや真の意味での処理能力の比較、ファイルの保護、ネットワーク、ユーティリティプログラムなどを含めた総合的な比較が必要である。商業的価値のなくなった旧式の計算機を安価に使用しつづけるのでは新しい概念を教育することはできない。今後は計算機関連の支出の大半がソフトウェアの費用である時代となるはずであり、十分な予算措置が必要である。

4. ま と め

当委員会では、現在の我が国の情報系専門学科の教育の問題点を指摘し、5年程度の将来を目標にモデルカリキュラムの案の作成にとりかかった。それまでは本提言のカリキュラム J 90 を暫定カリキュラムとして推奨することにした。カリキュラム J 90 の要件を満たしていない情報系専門学科は至急このレベルに達してほしい。

日本の学会が当該専門学科のアイデンティティを保ち卒業生の専門能力を明確にするためにモデルカリキュラムの検討に取り組んだのは画期的なことであり、カリキュラムの検討が継続されるとともに、その成果を専門学科へフィードバックできる体制を整えることが望まれる。

参 考 文 献

- 1) CURRICULUM Committee on Computer Science: Curriculum '68, Recommendations for Academic Programs in Computer Science, Comm. ACM, 11, pp. 151-197 (1968).
- 2) Austing, R., Barnes, B., Bonnette, D., Engel, G. and Stokes, G. (eds.): Curriculum '78—Recommendations for the Undergraduate Program in Computer Science, A report of the ACM Curriculum Committee on Computer Science, Comm. ACM, 22, pp. 147-166 (1979).
- 3) Denning, P. J., Comer, D. E., Gries, D., Mulder,

- M. C., Tucker, A. B., Turner, A. J. and Young, P. R.: *Computing as a Discipline*, Comm. ACM, 32, pp. 9-23 (1989). [邦訳 (木村 泉), 学問としての計算機分野, 情報処理, Vol. 31, No. 10, pp. 1351-1372 (1990)].
- 4) Tucker, A. B. et al.: *Computing Curricula 1991—Report of the ACM/IEEE-CS Joint Curriculum Task Force*, ACM (1990).
- 5) 野口正一, 中森眞理雄: 大学等における情報処理教育の諸問題—平成元年度の調査研究を中心として, 情報処理, Vol. 31, No. 10, pp. 1373-1389 (1990).
- 6) 情報処理学会大学等における情報処理教育検討委員会: 平成元年度教育改革の推進に関する研究委託中間報告書 (1990年3月).
- 7) 情報処理学会大学等における情報処理教育検討委員会: 大学等における情報処理教育のための調査研究報告書 (1991年3月).

付録 1 カリキュラム J 90

カリキュラム J 90 は 15 の授業科目からなる。このうち、コアカリキュラム (専門学科で教えるべき最小限のカリキュラム) は次の 7 科目である。情報系専門学科の学生はこのコアカリキュラムに対応する内容をすべて履修すべきである。

- JCS 1 プログラミング序論
- JCS 2 プログラムの設計と実現
- JCS 3 計算機システム序論
- JCS 4 計算機ハードウェア基礎
- JCS 5 情報構造とアルゴリズム解析
- JCS 6 オペレーティングシステムとアーキテクチャ I
- JCS 7 プログラミング言語の構造

次の JCS 8 から JCS 15 までは専門性がやや強く、情報系専門学科の学生はこれらの中から選択的に半分程度を履修し、特に 1 科目を深く履修することが望ましい。

- JCS 8 オペレーティングシステムとアーキテクチャ II
- JCS 9 ファイルとデータベースシステム
- JCS 10 人工知能
- JCS 11 ヒューマンインタフェース
- JCS 12 計算のモデルとアルゴリズム
- JCS 13 ソフトウェアの設計と開発
- JCS 14 プログラミング言語の理論と実際
- JCS 15 数値計算の理論と実際

カリキュラム J 90 の授業科目の間には、あらかじめ履修しておくべきことを指定した先修条件

がある。ただし、授業の進め方など運用を工夫すれば、先修条件が指定された授業科目同士を並行して開講することも可能である。カリキュラム J 90 の各授業科目 JCS 1~JCS 15 の目的・内容と先修条件は次のとおりである。

コアカリキュラム

情報系専門学科の学生は、コアカリキュラムを構成する次の JCS 1~JCS 7 をすべて履修すべきである。

JCS 1 プログラミング序論

本授業科目の目的は、

(1) 問題解決手法とアルゴリズム化を秩序だてて考える能力を養成すること

(2) プログラムの設計、コーディング、デバッグ、テスト、文書化について良質のプログラム書法を用いて習得させること

(3) 計算機科学をさらに学んでいくための基礎を習得させること

である。そのために、

(a) 簡単な例題を用いて、トップダウンにプログラムを作成させること。並行して、文書化を励行させること

(b) プログラムの動作を字面の上で追跡させること。プログラムの“状態”を考慮することが有用であることを認識させること

(c) 構造的プログラミングに適した高水準プログラミング言語を教えること。それに対応したプログラム図式化法を教えること

(d) 手続き抽象化とデータ抽象化の概念を教えること

(e) プログラミング環境やネットワーク環境についての基本的理解を得させること

が必要である。関連知識として、

(f) 計算機のハードウェアおよびソフトウェアの技術の発展について紹介すること

は、問題解決法、アルゴリズム、プログラミング手法なども固定したものではなく、計算機技術とともに発展してきたものであり今後も発展していくものであることを教えるうえで効果がある。

本授業科目の開始前に学生が履修しておくべき科目はないが、冒頭で (e) を主眼とした実習 (キーボードとタッチタイプ、エディタ、ファイルシステム、コマンド、メールなど) を課することが望ましい。

JCS 2 プログラムの設計と実現

先修条件: JCS 1

本授業科目の目的は、

(1) 計算機による一層の問題解決能力、信頼性が高く効率の良いプログラムを作成する能力を養うために、プログラミングの方法論とそれを支えるアルゴリズムを教育すること

(2) 計算機科学をさらに学んでいくための基礎を習得させること

である。そのために、

(a) 引き続き、構造的プログラミングに適した高水準言語を用いたプログラムの設計、コーディング、テストの手順を秩序だてて教えること

(b) データ抽象化の利用法を教えること。ここで使用するプログラミング言語があらかじめ提供している基本のデータ型以外のデータ構造（リンクされたリスト、スタック、待ち行列、木など）を、例題を用いて教えること。それらのデータ構造のさまざまな実現方法を理解させること

(c) 再帰の考え方を教えること

(d) アルゴリズムの複雑度（手間、効率）の概念とその評価のしかたを教えること。たとえば、探索（サーチ）と整列（ソート）のアルゴリズムとその解析の初歩を教えること

(e) プログラムの正しさの証明の初歩を教えること

(f) 数値を取り扱う際に生ずる誤差に関するセンスを養うこと
などが必要である。

JCS 1 と JCS 2 は続けて開講されるので、一体として扱ったり、内容を相互に入れ換えたりすることも可能である。たとえば、JCS 2 の(f)（誤差）は JCS 1 で教えることも可能である。また、JCS 2 (d)で探索と整列を教えるときは、データを整列しておくことにより探索の手間を大幅に減らすことができることを示して整列の意義を理解させるなど、JCS 1 の目的(1)、(2)を絶えず強調することが大切である。

JCS 3 計算機システム序論

先修条件: JCS 2

本授業科目の目的は、

(1) 計算機システムの基本概念を機械語のレベルで習得させること

(2) 計算機アーキテクチャの基本概念を習得

させること

である。そのために、

(a) フォンノイマン計算機の仕組み（CPUアーキテクチャ）を教えること

(b) コード、アドレス、数の表現などの基本的概念を教えること

(c) アセンブリ言語によるプログラミングを教えること。サブルーチン、コルーチン、再入可能ルーチンなどの概念を教えること

(d) 割り込みの概念、および各種の入出力装置と CPU との関係について教えること

(e) アセンブリ言語と高水準言語との対応づけを教えること

などが必要である。なお、(c)(アセンブリ言語)と(d)(割り込み)については、実在する（実在した）計算機のアセンブリ言語による十分な量の演習が不可欠である。できれば、オペレーティングシステムが存在を前提としないで、イニシャルプログラムローダ、アセンブラ（1パス、2パスなど）、リロケータブルローダを作成させることが望ましい。

JCS 4 計算機ハードウェア基礎

先修条件: JCS 2

本授業科目の目的は、

(1) 計算機の主要な構成要素の構造や仕組みなど、ハードウェアの基本概念を習得させること

(2) 新しいアーキテクチャに基づく計算機を設計したり実現したりすることができるように、ハードウェアのレベルでのアルゴリズムの考え方を習得させること

である。そのために、

(a) ゲート、マイクロプログラム、機械語、アセンブリ言語、オペレーティングシステムマクロ、高水準言語の各レベルにおけるアーキテクチャの概念を教えること

(b) 順序機械の表現・簡約化・状態割当・実現・テスト、非同同期順序機械を教えること

(c) 集積回路（IC、VLSI、ULSI など）のモデル、ゲートのレベルでの計算複雑度の概念を教えること

などが必要である。本授業科目の本質は、装置や回路の特性、半導体の物性などだけでなく、アルゴリズムとアーキテクチャのレベルでハードウェアを論ずることである。

JCS 5 情報構造とアルゴリズム解析

先修条件: JCS 3, JCS 4

本授業科目の目的は、

(1) 各種のデータ構造について、さまざまな非数値的アルゴリズムを解析し、設計に用いる方法を習得させること

(2) データを扱う際の手法の選択という問題に対して、アルゴリズム解析に基づいた設計の規範を作れるように指導すること

(3) システムプログラムで用いられる構文解析や記憶域管理などの基本的アルゴリズムを習得させること

である。そのために、

(a) スタック、待ち行列、リスト、木のような基本的なデータ構造を教えるとともに、データ抽象化の考え方を教えること。それらのデータ構造を、リンクされたリストや配列を用いて実現する各種の方法を教え、時間と記憶場所の大きさなどに関するトレードオフを比較・検討させること

(b) グラフ理論の初歩をアルゴリズムの観点から教え、基本的アルゴリズム（経路探索、スパニング木生成など）を教えること

(c) 内部記憶や外部記憶装置を想定した整列、併合（マージ）、探索のアルゴリズムを通じて、複雑度の概念の現実的意味を把握させること

(d) ハッシュ、動的記憶割り付け、ガベージコレクションなどの記憶域管理アルゴリズムを教えること

(e) 簡単な構文解析アルゴリズムとその実現方法を教えること
などが必要である。

JCS 6 オペレーティングシステムと**アーキテクチャ I**

先修条件: JCS 3, JCS 4

本授業科目の目的は

(1) CPU 命令セットアーキテクチャの概念、オペレーティングシステムの役割と概念、オペレーティングシステムとアーキテクチャとの関係(OS アーキテクチャ)を理解させ、オペレーティングシステムを設計する規範を習得させること

(2) 多様なレベル・形態の計算機結合方式(バス、ネットワークなど)を含むシステムを構築する規範を習得させること

である。そのために、

(a) オペレーティングシステムの役割(資源管理、仮想化)を定義し、オペレーティングシステムの設計がユーザやプログラムと計算機とのインタフェースを決めることであることを理解させること

(b) オペレーティングシステムからみたハードウェアについて教えること

CPU アーキテクチャとオペレーティングシステムとのインタフェースを明らかにすること

(c) プロセス、スケジューリング、同期などの概念と実現方式を教えること

(d) 入出力管理、主記憶管理、ファイル管理、通信管理などについて教えること

(e) ユーティリティライブラリを構成するための技法を教えること

などが必要である。

JCS 7 プログラミング言語の構造

先修条件: JCS 3, JCS 4

本授業科目の目的は、

(1) プログラミング言語の仕組み、特に、実行時のプログラムの振舞いについて理解させること

(2) プログラミング言語を設計するための理論的基礎を習得させること

(3) 言語処理系(コンパイラやインタプリタなど)を作るための基礎的能力・素養を養成すること

である。そのために、

(a) データの型と構造体、制御構造とデータフローを教えること

(b) 名前の有効範囲、引数の受け渡し(値渡し、変数渡し)、再入可能プログラム、静的変数と動的変数、副作用などの概念を教えること

(c) 現存の各種のプログラミングスタイルの概観を与えること。たとえば、手続き型プログラミング(Pascal, Modula-2, C など)のほかに関数型プログラミング(Lisp など)、論理型プログラミング(Prolog など)、オブジェクト指向プログラミング(Smalltalk, C++ など)などとそれらの応用例を教えること

(d) 言語処理系の形態(コンパイラ、インタプリタ)と原理を教えること

などが必要である。

以上の7科目をコアカリキュラムとする。情報

系専門学科の学生なら、このコアカリキュラムはすべて履修しておくべきである。

上級科目

次の JCS 8 から JCS 15 までの 8 科目は上級科目であり、情報系専門学科の学生がすべてを履修する必要はないが、少なくとも半分程度は履修し特に 1 科目を深く研究することが望ましい。

JCS 8 オペレーティングシステムと

アーキテクチャ II

先修条件: JCS 6

本授業科目の目的は、

(1) ハードウェアの上にオペレーティングシステムを設計したり、新しいアーキテクチャを設計したりする能力を養成すること

(2) 分散処理システムを構築する能力を養成すること

(3) システムの性能を評価するためのモデル化と解析の手法を習得させること

である。教えるべき内容は、たとえば、

計算の保安性、認証、並列処理、デッドロック
分散処理、クライアントサーバモデル、ネットワーク、階層プロトコル

性能評価、ユーティリティライブラリ

JCS 9 ファイルとデータベースシステム

先修条件: JCS 5, JCS 6

本授業科目の目的は、

(1) 2 次記憶装置に大量の情報を蓄え検索に供するシステムを設計したり実現したりするための規範を習得させること

(2) 情報の物理的構造と論理的構造の違いを認識させ、その違いを支える仕掛の原理を理解させること

である。教えるべき内容は、たとえば、

ファイルシステム、ファイルの構成要素、
ファイルの操作

2 次記憶装置とソフトウェアインタフェース
データベースのモデル、データベースの操作、
データベースの検索

関係代数、依存性の理論、多重化記録の同時更新と一貫性

ハイパテキスト、図形ベース、イメージベース、知識ベース

JCS 10 人工知能

先修条件: JCS 5

本授業科目の目的は、

(1) 人間の知能と行動のモデルを記述する方法やそれらをシミュレートする計算機の究極的性能を論ずる価値基準や評価尺度を体得させること

(2) 知覚、運動、学習の構造を調べるための指針を学ばせること

(3) 未知の領域を開拓し新しい問題解決手法を編み出す能力を養成すること

である。教えるべき内容は、たとえば、

知識の表現 (論理、ルール、フレームなど)

と処理方式 (演繹、推論)

自然言語の理解と表現、機械翻訳、音声の認識と合成、ニューロン回路網

論理プログラミング、定理の自動証明、推理と学習、認知

ゲーム、探索手法、ロボット

JCS 11 ヒューマンインタフェース

先修条件: JCS 5, JCS 6

本授業科目の目的は、

(1) 人間から計算機へ入力する方法、計算機から人間へ出力する方法を、誤認識の防止や効率という観点から論じ、より良いヒューマンインタフェースを設計する指針を習得させること

(2) 対象物を理解しやすいように視覚化するためのグラフィックスの手法を習得させること

である。教えるべき内容は、たとえば、

計算幾何学、色彩理論

利用者の認知モデル

図形表示アルゴリズム (隠れ線、光線追跡、
スプライン曲線、テクスチャなど)

画像強調、CAD のためのモデル、グラフィックスアルゴリズム

日本語インタフェース

インタフェースの評価技法、各種対話技法の比較評価

本授業科目は歴史の浅い学問分野であることから、大学院の授業科目とするのが適当な場合もある。

JCS 12 計算のモデルとアルゴリズム

先修条件: JCS 5, JCS 6, JCS 7

本授業科目の目的は、

(1) 将来のアーキテクチャに対応して、アルゴリズムを解析したり考案したりすることができ
る能力を養成すること

(2) 形式言語の理論を通じて、文法やアルゴリズムの複雑さの階層を理解させること

(3) 計算機にできることとできないことの境界を見極めさせること

(4) 計算の複雑度に基づいたアルゴリズムや問題の分類の理論を教え、問題の現実的な意味での難しさを捉える能力を養成すること

(5) アルゴリズムを設計する際の基本的手法を習得させること

である。教えるべき内容は、たとえば、

各種計算モデル（ゲート、オートマトン、チューリング機械、ランダムアクセス機械、並列計算機など）に対応した計算の複雑度、 O 記法

形式言語、 λ 計算、万能チューリング機械、計算可能性、停止問題

多項式オーダーのアルゴリズム、非決定性アルゴリズム、NP 完全性

アルゴリズム設計の一般的方法（分割統治法、動的計画法、貪欲算法、分枝限定法など）

JCS 13 ソフトウェアの設計と開発

先修条件：JCS 5, JCS 7

本授業科目の目的は、

(1) 大規模ソフトウェアを多人数で開発したり保守したりするための技術や管理の方法を紹介すること

である。教えるべき内容は、たとえば、

プログラムの検証と証明、要求仕様言語、プログラムのテスト法

時制論理、抽象データ型、段階的詳細化、分割コンパイル、情報隠蔽

プログラム開発環境、プロジェクト管理、プロファイル作成、文書整形

ソフトウェアライフサイクル、ソフトウェアの信頼性

本授業科目は、大きなプログラムを作ったり複数人でプログラムを作ったりした経験の乏しい学生には理解不可能である。大学院の授業科目とするのが適当な場合もあろう。

JCS 14 プログラミング言語の理論と実際

先修条件：JCS 7

本授業科目の目的は、

(1) 言語処理系を作る際の基本的手法を習得させること

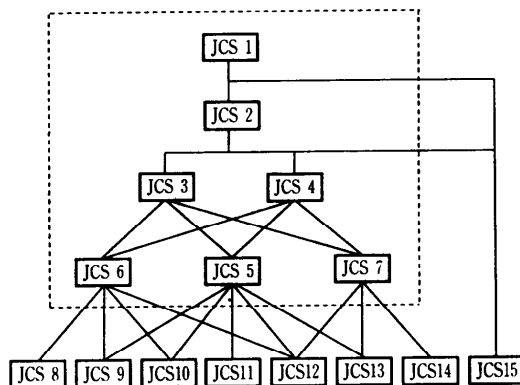


図-1 カリキュラム J90 の各授業科目間の先修条件

である。教えるべき内容は、たとえば、

語彙解析、構文解析、型検査、レジスタ割り付け、コード生成

最適化、並列性の検出、構文的・意味的な誤り検査

形式的意味論、パーザ・スキャナのジェネレータ

JCS 15 数値計算の理論と実際

先修条件：JCS 1, JCS 2

本授業科目の目的は、

(1) 連続量で記述されるシステムを離散量でできるだけ精度良く近似するための方法を習得させること

(2) 精度と計算時間とのトレードオフに関するセンスを養うこと

である。教えるべき内容は、たとえば、

浮動小数点方式、関数の多項式や有理関数による近似、丸め誤差解析の方法

行列計算、固有値問題、線形計画問題、非線形方程式の根

数値積分、常微分方程式の数値解法、偏微分方程式の数値解法

繰り返し解法と数値的収束、加速

付録 2 デニング図を用いた

カリキュラム評価法*

ACM 88 提言によれば計算機分野は次の副領域に分割される。

1. アルゴリズムとデータ構造
2. プログラミング言語

* 付録 2 の記述は ACM 88 提言²⁾の翻訳である「学問としての計算機分野」(木村泉訳、情報処理、第 31 巻、第 10 号、pp. 1351-1372)によるものである。

3. アーキテクチャ
4. 数値的および記号的計算
5. オペレーティングシステム
6. ソフトウェア方法論とソフトウェア工学
7. データベースと情報検索
8. 人工知能とロボティクス
9. 人とコンピュータのコミュニケーション

また、同提言によれば計算機分野では、理論、抽象化、設計の三つのパラダイムがともに重要である。

第1のパラダイムである理論とは、

- (1) 研究の対象を特徴づける (定義)
- (2) 対象間に存在すると思われる関係について仮説を立てる (定理)
- (3) それらの関係が確かに存在するかどうか確かめる (証明)

- (4) 結果を解釈する

の4段階を経て、首尾一貫し、筋の通った理論へと導くものである。

第2のパラダイムである抽象化 (モデリングとか実験とか呼んだほうが適切な可能性もあろう) とは、

- (1) 仮説を形成する
- (2) モデルを構成し、予測を立てる
- (3) 実験を設計し、データを集める
- (4) 結果を解析する

の4段階を経て、現象の解明に導くものである。

第3のパラダイムである設計とは、

- (1) 要求を述べる
- (2) 仕様を述べる
- (3) システムを設計、製作する
- (4) システムをテストする

の4段階を経て、与えられた問題を解くシステムないし装置の構築に導くものである。なお、上記の三つのパラダイムにはそれぞれ4つの段階があるが、誤りや整合性の欠如、モデルの予測と実験結果との不一致、システムが要求を十分に満たしていないことなどが分かった場合にはそれらの4段階が繰り返されることは言うまでもない。

計算機分野は、これら三つのパラダイムが交差する位置にあるという意味で、理工学やその他あらゆる学問分野の中で特に際だった性格を有している。これらのパラダイムはどれが一番基本的であるかを論ずることには意味がない。一方、これ

ら三つのパラダイムは、職業能力の異なる領域を示しているという意味では異なるものである。

これらの副領域とパラダイムに基づいて、計算機分野に対する次の 9×3 の定義マトリクス (デニング図) が構成される。

現実には、各授業科目、各研究者、各研究論文がデニング図の27個の枠のいずれか一つに属するということではまれであり、多くの場合は複数の枠にまたがるであろう。すなわち、デニング図は「分布図」として使われるのが普通であろう。

ACM 88 提言におけるデニング図とは趣旨が異なるが、情報系専門学科のカリキュラムをデニング図にあてはめる (すなわち、各授業科目に対するデニング図を重ね合わせる) ことも、カリキュラムの総合的な評価という点で、意味がある。実際、米国には、この方法でカリキュラムを自己評価している大学がある^{*}。デニング図が唯一無二で最良の方法とは言えないが、共通の尺度でカリキュラムを比較する便利な方法であるので、広く活用されることが望まれる。

もちろん、デニング図を完全に覆うカリキュラムが優れているというわけではない。デニング図を100% 覆うのは、授業時間数やスタッフの数の制約から、不可能であろう。デニング図の覆い方の片寄りはその学科の特徴を表すとも言えよう。しかし、デニング図をきわめてわずかしき覆うことができないカリキュラムは情報系専門学科のカリキュラムとしては不十分と考えられる。

各情報系専門学科とも、デニング図を用いてカリキュラムの自己評価を試みることを推奨する。

	理 論	抽 象 化	設 計
1. アルゴリズムとデータ構造			
2. プログラミング言語			
3. アーキテクチャ			
4. 数値的および記号的計算			
5. オペレーティングシステム			
6. ソフトウェア方法論とソフトウェア工学			
7. データベースと情報検索			
8. 人工知能とロボティクス			
9. 人とコンピュータのコミュニケーション			

図-2 デニング図

^{*} たとえば、ニューヨーク州立大学バッファロー校。

付録 3 実験・演習の例, 試験問題の例

問題例 1

ALGOL 言語に関する 1960 年版の報告書には, 次のような構文の定義が与えられていた.

```

<statement> ::= <unconditional statement> |
               <conditional statement>
<unconditional statement> ::= <basic statement> |
               <for statement> |
               <compound statement> | <block>
<conditional statement> ::= <if statement> |
               <if statement> else <statement>
<if statement> ::= <if clause>
               <unconditional statement> |
               <label> : <if statement>
<if clause> ::= if <Boolean expression> then
<for statement> ::= <for clause> <statement> |
               <label> : <for statement>
しかるに 1963 年版の改訂報告書では, 第 1, 2,
3, 4 式が
<statement> ::= <unconditional statement> |
               <conditional statement> |
               <for statement>
<unconditional statement> ::= <basic statement> |
               <compound statement> | <block>
<conditional statement> ::= <if statement> |
               <if statement> else <statement> |
               <if clause> <for statement> |
               <label> : <conditional statement>
<if statement> ::= <if clause>
               <unconditional statement>

```

と変わっている.

(a) この改訂で解決された問題点はどんなものであったか. 具体例をあげて説明せよ.

(b) 同じ問題点を解決するこれ以外の方法としてどんなものが考えられるか. その利害得失はどうか.

問題例 2

たいていの計算機には, サブルーチン呼び出し用の命令が用意されており, CALL 命令, BAL 命令, JSR 命令, JST 命令などと呼ばれている. これらの命令は, リンク情報(サブルーチンの実行終了後に, 呼び出し以前の状態を復元するのに必要な情報)を確保したうえでサブルーチンへの飛

び越しを起こす, という点では共通しているが, リンク情報のしまい場所についてはさまざまであり, たとえば次のようなやりかたが行われている.

a. リンク情報をサブルーチンの先頭番地以下に格納し, その直後から実行をはじめる.

b. リンク情報を汎用レジスタのどれかに確保する.

c. スタックポインタと称する特別のレジスタを用意し, リンク情報はそのレジスタの内容で指定された記憶場所にしまう. なおその際, スタックポインタの値を, 次回には隣の記憶場所が使われることになるように, 自動的に調整する.

設計 a, b, c にはそれぞれどんな長所短所があるか, 述べよ.

問題例 3

ある小さな Pascal プログラムの中で, 2 次元配列 A が次のように定義されているとする.

```

var A : array [1..1000] of array [1..1000]
      of integer ;

```

またこのプログラムの中では, A を単位行列にするために次のような操作が行われているという.

```

for i:=1 to 1000 do
  for j:=1 to 1000 do
    if i=j then
      A[i][j] := 1
    else
      A[i][j] := 0 ;

```

} ...操作 1

(a) 整数の比較, 変数や配列要素への代入に, それぞれ 1μ 秒の CPU 時間を要するものとし, 全比較時間と全代入時間の和を総 CPU 時間と名づける (比較および代入以外の操作に要する時間は無視する). ただし for 文は毎回の繰り返しおよび脱出時に各 1 回の代入と比較を行うものとする. そのとき,

(a1) 操作 1 に要する総 CPU 時間を秒の単位で計算せよ. 秒未満は切り捨てとし, 計算の筋道を簡単に示せ. 以下同様とする.

(a2) 総 CPU 時間を短縮するために, 操作 1 を下記の操作 2 のように改良した. 空白部分 (イ) および (ロ) に適当な文を埋め込み, 操作 2 の総 CPU 時間を計算せよ.

```

for i:=1 to 1000 do
  for j:=1 to 1000 do
    (イ) ;
  for i:=1 to 1000 do
    (ロ) ;
  } ...操作 2

```

(b) このプログラムは、ある LRU (Least Recently Used) 置き換えによるデマンドページング方式のメモリ管理を行っているシステムの上で走行しているという。各ページは 2000 語で、このプログラムにはページ枠が 10 個割り当てられているとする。ただし 1 語は 1 番地で、1 語に整数値が 1 個入る。コンパイルされたプログラムと、配列以外の変数および環境データは仮想空間上のページ 0 に割り当てられ、また A[1][1] はページ 1 の先頭番地に割り当てられる。上記の操作の実行を開始する時点では、ページ 0 はすでにあるページ枠に格納されており、他の 9 個のページ枠は空であるとする。1 回のページフォールトの処理には 10 m 秒を要するものとし、総 CPU 時間と全ページフォールト処理時間の和を総実行時間と名づける。そのとき、

(b1) 操作 1, 操作 2 についてそれぞれ総実行時間を求めよ。ただし他のプログラムによる影響はないものと仮定してよい。

(b2) 操作 1, 操作 2 より総実行時間を短くするために、プログラムを下記の操作 3 のように変更した。空白の部分を補い、その場合の総実行時間を求めよ。

```

for i:=1 to 1000 do
  begin
  ...
  end;
} ...操作 3

```

問題例 4

印刷用紙上に文字を打ち並べることによって線画を描くためのプログラムについて、下記に答えよ。差し当たっては、たとえば入力として

2, 6—2, 29—9, 22—17, 4; 9, 22—17, 30.

のようなものを与えれば、印刷用紙上に数が打ち出される、という程度のものを考える。すなわち 2 行目の 6 文字目から 2 行目の 29 文字目、9 行目の 22 文字目を経て 17 行目の 4 文字目までをこの順に結ぶ線分群と、9 行目の 22 文字目から 17 行目の 30 文字目までを結ぶ線分が、たとえば単価記号を打ち出すことによって描かれ、全体としてはカタカナの「ス」の字ができる、というよ

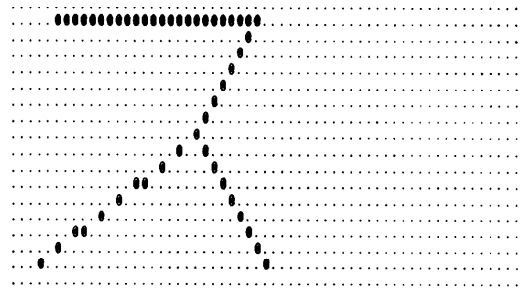


図-3

うにする。図では字数を数えやすくするため、背景をピリオドで満たしたが、これは実際には空白で置き換えられるものとする。

(a) このプログラムの利用者から将来出ると予想される機能向上の要求にはどんなものがあるか。

(b) それらの機能向上を無理なく実施できるようにするためには、プログラムはどんな機能単位に分解しておくのがよいと思うか。またそう思う理由は何か。

問題例 5

データベースの設計問題を考える。扱うべき属性には、次のものがあるとする。

工場、所在地、製品、重量、定価、協力工場
これらの間には次に示す関連がある。

- 工場の所在地は、所在地で示される。
- 重量、定価は、製品に対応するものである。
- 各工場は、いくつかの製品を作っている。
- 工場ごとに、いくつかの協力工場が決まっている。

この場合にどのような従属性制約が考えられるか。これらの従属性制約を用いた関係データベースの設計を行え。次に、設計した関係集合に対し、ある協力工場を与えて、それが協力している工場（複数あり得る）で作られている製品のうち、重量が 10 kg 以下のものをすべて求める質問を書け。また、それを効率よく処理するにはどうすればよいか述べよ。

大学等における情報系専門教育の改善への提言の発表について（付記）

情報処理学会は文部省高等教育局専門教育課の「平成元年度教育改革の推進に関する研究委託」による「大学等における情報処理教育の改善のた

めの調査研究」を実施するために、学会内に「大学等における情報処理教育検討委員会」（野口正一委員長）を設け、2年間、調査研究を実施した。委員会のメンバと活動は参考文献5)に報告されているとおりである。

委員会は、実質的な審議を深く行うために、CS（コンピュータサイエンス）、IS（情報システム）、一般情報処理教育の三つの分科会を設けた。CS分科会は情報工学科・情報科学科などの教育を対象とし、IS分科会は情報システムを構築するシステムエンジニアを養成する学科の教育を対象とし、一般情報処理教育分科会は理工系学部の非情報系学科や非理工系学部の情報処理教育を対象としている。本提言はCS分科会の作業の成果であり、平成3年3月20日の大学等における情報処理教育検討委員会で承認を受けたものである。

本学会に限らず、我が国の学会が大学のカリキュラムについて組織的に研究したということは、前例がない。このような検討の意義と必要性を認めた文部省と本学会理事会の先見の明は特筆するに値する。米国のACM（Association for Computing Machinery）では20年以上のカリキュラム検討の歴史がある。本提言で、「暫定」とはいえモデルカリキュラムJ90を提案したことにより、本学会とACMが共通の土俵に上る最低限の素地ができたことになる。J90はACMのこれまでのカリキュラム68¹⁾、カリキュラム78²⁾や提言88³⁾を大いに参考にしているが、たとえば、アルゴリズム論の基礎を扱うJCS5やJCS12のように、ACM案より目的と方向が一層明確になっていることが理解されよう。

一方、大学教育に関する考え方には日米で違いもあり、このことは、今後、本学会が暫定ではない長期的なカリキュラムを策定していく過程ではっきりしてくると思われる。いずれにしても、日米両国が情報処理教育カリキュラムに関して良い意味で競争することは人類の発展と幸福につながるはずであり、大いに期待していただきたいと思う。

カリキュラムというものは4年間という限られた年限の中に教えるべきことがらを組み入れるものであり、すべてを盛り込むことは不可能である。特に、J90の「最小限」という性格上、策定

にあたっては、「何を盛り込まないか」が当CS分科会の議論的になった。自分が論文を書いたことがある分野がわずかに1語しか触れられていない（あるいは、まったく触れられていない）という不満を感じる読者も多いと想像される。これは教育と研究の差であり、J90に「偏り」が感じられるとすればそれは教育的配慮によるものと解釈されたい。また、J90のコア部分JCS1～JCS7をすべて教えても20～30単位程度であり、理工系の学生が卒業までに取得する専門科目の総単位数の半分にも満たない。実際のカリキュラム編成にあたっては「理工系学問の一般的素養」や「各大学の個性」と調和させつつ充実を図る必要がある。いずれにしろ、カリキュラムJ90を排他的性格のものとは理解しないでいただきたい。

大学等における情報処理教育検討委員会は平成3年3月で期限を終了したが、カリキュラム検討は「情報処理教育調査委員会」で継続されることになっている。

謝辞 大学等における情報処理教育検討委員会の市川照久委員（三菱電機(株)）には、平成元年度のカリキュラム検討委員としてカリキュラム検討作業に参画していただいた。御牧義委員（電気通信大学）には、ほとんどCS分科会委員と同様に活動していただいた。堂下修司委員（正幹事、京都大学）、有澤博委員（横浜国立大学）、乾信委員（長岡技術科学大学）、大岩元委員（豊橋技術科学大学）、川合慧委員（東京大学）、藤野喜一委員（日本電気(株)）、村岡洋一委員（早稲田大学）には合宿を含む検討会に参加していただいた。國井利拳委員（東京大学）には有益な助言をいただいたうえに資料の収集に関してもお世話になった。以上の各委員に感謝申しあげます。

本提言を当学会誌に掲載することを快諾された大学等における情報処理教育検討委員会の全委員、および文部省高等教育局専門教育課に感謝申しあげます。

（平成3年6月14日受付）

**野口 正一** (正会員)

昭和5年生。昭和29年東北大学工学部電気工学科卒業。昭和35年同大学院博士課程修了。工学博士。昭和46年東北大学電気通信研究所教授。昭和59年東北大学大型計算機センター長。平成2年東北大学応用情報学研究センター長。主として情報システム構成論、知識処理に関する研究に従事。著書「情報ネットワーク理論」(岩波)、「知識工学基礎論」(オーム社)など。

**高橋 延匡** (正会員)

昭和8年生。昭和32年早稲田大学第一理工学部数学卒業。同年(株)日立製作所中央研究所入社。HITAC 5020 モニタ, TSS の開発に従事。昭和52年より東京農工大学工学部数理解情報教授。平成元年電子情報教授。理学博士。オペレーティングシステム, 日本語情報処理, パターン認識の研究に従事。電子情報通信学会, ソフトウェア科学会, 計量国語学会, ACM 各会員。

**牛島 和夫** (正会員)

1937年生。1961年東京大学工学部応用物理学科(数理工学)卒業。1963年同大学院修士課程修了。同年九州大学中央計数施設勤務。1977年九州大学工学部情報工学科教授(計算機ソフトウェア講座担当)。現在に至る。1990年4月から九州大学大型計算機センター長を兼務。工学博士。日本ソフトウェア科学会, 電子情報通信学会, ACM 各会員。

**都倉 信樹** (正会員)

昭和38年大阪大学工学部電子工学科卒業。昭和43年同大学院博士課程修了。同年大阪大学基礎工学部講師。昭和52年同教授(情報工学科)。主として, 計算機言語, ソフトウェア, VLSI アルゴリズムに関する研究に従事。電子情報通信学会, 日本ソフトウェア科学会, 人工知能学会, ACM 各会員。

**榎本 彦衛** (正会員)

昭和20年生。昭和43年東京大学理学部数学科卒業。45年同大学院修士課程修了。同年京都大学理学部数学科助手。現在, 慶應義塾大学理工学部数理科学科教授。離散数学, 特にグラフ理論に関する研究に従事。理学博士。日本数学会, 日本ソフトウェア科学会, 日本応用数理学会各会員。

**諸橋 正幸** (正会員)

1948年生。1972年早稲田大学理工学部電気工学科卒業。1974年同大学院理工学研究科電気工学修士課程修了。同年日本アイ・ビー・エム(株)入社。東京基礎研究所に勤務。自然言語処理に関する研究開発に従事。電子通信情報学会, 日本ソフトウェア科学会などの会員。

**木村 泉** (正会員)

1935年生。1960年東京大学理学部物理学科卒業。1965年同大学院博士課程退学。東京大学助手。東京教育大学講師を経て, 現在東京工業大学教授。計算機システムのヒューマンインタフェースに関する研究に従事。IEEE, ACM, 電子情報通信学会, ソフトウェア科学会, Human Factors Society 各会員。

**中森真理雄** (正会員)

昭和46年東京大学工学部計数工学科卒業。昭和52年同大学院修了。工学博士。同年東京農工大学工学部数理解情報講師。現在教授。昭和60年度文部省在外研究員として西ドイツ, ボン大学で研究。密合せの数値計画問題, グラフ・ネットワークフロー, アルゴリズム・データ構造, 地理情報処理, 集積回路の配置配線設計, CAI, ソフトウェア開発手法などの研究と教育に従事。