

## Java 言語の理解順序に関する関連構造の統計的分析

武村 泰宏<sup>†\*</sup> 島 和之<sup>†</sup> 松本 健一<sup>†</sup> 井上 克郎<sup>†‡</sup>

<sup>†</sup> 奈良先端科学技術大学院大学 情報科学研究科

〒630-0101 奈良県生駒市高山町 8916-5

Phone: 07437-2-5312, Fax: 07437-2-5319

<sup>‡</sup> 大阪大学 大学院基礎工学研究科

〒560-8531 大阪府豊中市待兼山町 1-3

※大阪芸術大学短期大学部

〒546-0023 大阪市東住吉区矢田 2-14-19

E-mail {yasuhi-t, shima, matumoto}@is.aist-nara.ac.jp, inoue@ics.es.osaka-u.ac.jp

あらまし 本研究では、Java 言語の理解順序がどのようになっているかを、学習者の理解状態をテストによって測定し、順序理論の依存関係に矛盾率の概念を取り入れた統計的分析手法を提案して解明した。依存関係は、ある項目の理解に他の項目の理解を必要条件とする知識の関連で、全体の依存関係の構造が関連構造である。矛盾率は、ある条件のもとで依存関係に矛盾した集合に回答が含まれる確率で、二項分布の分布関数に偶然の成功と失敗の確率を適用した。矛盾率がある定数より小さければ依存関係が存在するが、知識項目だけの複雑な関連構造では理解順序を解明できない。そこで、因子負荷量から 25 個の知識項目を 9 つのクラスタに集約した関連構造に変換し、この関連構造を単純化して Java 言語の理解順序が把握できる簡潔な関連構造を描画した。

キーワード Java 言語, 依存関係, 学習者モデル, プログラミング教育

### A Statistical Analysis of the Relational Structure on the Sequence of Understanding the Java Programming Language

Yasuhiro TAKEMURA<sup>†\*</sup>, Kazuyuki SHIMA<sup>†</sup>, Ken-ichi MATSUMOTO<sup>†</sup>, and Katsuro INOUE<sup>†</sup>

<sup>†</sup> Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama, Ikoma, Nara 630-0101, Japan

Phone: 07437-2-5312, Fax: 07437-2-5319

<sup>‡</sup> Graduate School of Engineering Science, Osaka University

1-3 Machikaneyama-cho, Toyonaka, Osaka 560-8535, Japan

※Osaka University of Arts Junior College

2-14-19 Yata, Higashi-sumiyosi ku, Osaka 546-0023, Japan

E-mail {yasuhi-t, shima, matumoto}@is.aist-nara.ac.jp, inoue@ics.es.osaka-u.ac.jp

**Abstract** This research, which proposed taking in a concept of inconsistency rate in dependence relation of the Ordering theory, clarified sequence of understanding learners by measuring understanding state of Java programming language. Inconsistency rate signified probability that combinations of two answers based on certain condition were included in a set. In the set, inconsistency that mistaken answer was given to the easy test item and correct answer was given to the difficult test item, accrued in dependence relation of the Ordering theory.

**Keywords** Java programming language, dependence relation, learner's model, programming learning and instruction

## 1. はじめに

プログラム教育を系統的に体系化するには、学習フェーズにおける知識項目の理解の変化や、プログラム関連知識とソースコードとの関連などからの理解状態の解明が必要である。プログラム教育の特徴として、概念の間違ひは、一組の問題相互間の関係によって生じる。したがって、その間違ひやバグの項目を見つけるよりも、それに対応した要素の関連に着目することが重要である。また、ソフトウェア工学教育の一つであるプログラム教育を充実させるためには、実際の学習履歴の記録や、学習者を被験者とした実験を行い、そのデータを多面的に解析し、その結果をもとにしたプログラム教育の系統的な体系化が必要である。

学習者の思考、学習、知識獲得といった過程のモデル化では、(1) 知識表現 (プロダクション・ルール / フレーム) [11], (2) 算数の文書題における因果タイプ (単体変化問題 / 相対変化問題 / 比較問題 / 部分全体問題 / 時間問題 / 空間問題) [4], (3) 振る舞いに基づくプログラミングモデル [5] などが提案されている。また、プログラム言語の学習に関する研究もさまざまなアプローチで進められている。長谷川らは、初心者のプログラミング学習を対象に、プログラム制御構造のイメージと理解度について調査し、その回帰関係に関する研究を行った [2]。ここでは、オブジェクト指向、プログラム言語の関連知識やプログラムに関する理解との関連ではなく、プログラムの制御構造だけが対象にされている。竹谷らは教師が構築した認知マップを基準として、学習者がそれぞれの理解内容に基づき描画した認知マップを比較・分析し、学習者の認知・理解の程度を評価する形式的評価方法を提案している [9]。ここでは、知識項目の関連構造として表される認知マップは、教師や学習者自身の理解状態が主観的に描画されるものである。

本研究では、Java 言語におけるメソッド、クラス、インタフェースなどの概念とプログラムコードなどの知識項目の依存関係に注目した。依存関係は、ある知識項目の理解に他の知識項目の理解を必要条件としている知識項目間の関連である。学習者の理解順序がどのような構造になっているかを、学習者の理解状態を測定し、順序理論の依存関係に矛盾率の概念を提案した分析手法によって解明した。プログラム言語の教育において、学習者の知識項目の理解順序は、教育の体系化や知的教育システムの構築にとって重要であるが、学習者へのアンケートやインタビューでは、学習者自身の「誤った理解」や「知

識レベルの差異」などが影響して、理解状態を定量的に把握できない。またテスト結果の基礎的な統計処理だけでは、知識項目間の依存関係が見出せない。

学習者に Java 言語に関する講義と演習後にテストを行い、順序理論にもとづいてテスト結果を分析するが、「まぐれ当たり」や「うっかりミス」といった事象が、知識項目間の依存の強さに影響を及ぼす。また、知識項目の数が多くなると、知識項目の依存関係によって描かれた関連構造が複雑になり、学習者の理解順序を把握しにくい。

順序理論の依存関係において、「まぐれ当たり」と「うっかりミス」が及ぼす影響を小さくするため、二項分布の分布関数に偶然の成功と失敗の確率を適用した矛盾率によって、知識項目の依存関係を解析した。矛盾率は、2 つの回答の組み合わせがある条件のもとで、順序理論の依存関係に矛盾する集合に含まれる確率である。依存関係に矛盾した 2 つの回答は、易しいテスト項目が間違われ、難しいテスト項目を正解する組み合わせになる。「まぐれ当たり」は、多岐選択方式 (multiple-choice method) における偶然の成功の確率を適用した。「うっかりミス」については、Java 言語の 3 つの学習フェーズにおける理解状態の測定データ [8] から、偶然の失敗が生じる確率を求めた。因子負荷量から 25 個の知識項目を 9 つのクラスタに集約し、複雑な知識項目だけの依存関係をクラスタ間の依存関係へ変換する方法と、その関連構造の簡単化を提案した。この提案手法により、クラスタの簡潔な関連構造を描画して学習者の理解順序を解明した。

## 2. 知識項目の関連構造に関する研究

伊藤らは、関連構造を分析するため項目間の依存関係に注目し、Airasian & Bart の順序理論 [1] によって依存定数  $\epsilon$  を求めた [3]。順序理論の考え方は、ある学習集団の全体的傾向として、難しいテスト項目 Y の達成には、易しいテスト項目 X の達成を必要条件としている。テスト項目 X と Y の達成を "correct", 非達成を "error" とすると、n 人の学習者は以下の  $M_{00}, M_{01}, M_{10}, M_{11}$  のいずれかの集合に割り当てられる。

- ① 集合  $M_{00}$  は、テスト項目 X が "error", テスト項目 Y も "error" になる学習者の集合で、その学習者数は  $N_{00}$  になる。
- ② 集合  $M_{01}$  は、テスト項目 X が "error", テスト項目 Y が "correct" になる学習者の集合で、その学習者数は  $N_{01}$  になる。
- ③ 集合  $M_{10}$  は、テスト項目 X が "correct", テス

ト項目 Y が "error" になる学習者の集合で、その学習者数は  $N_{10}$  になる。

- ④ 集合  $M_{11}$  は、テスト項目 X が "correct"、テスト項目 Y も "correct" になる学習者の集合で、その学習者数は  $N_{11}$  になる。

テスト項目 X と Y の達成 "correct" と非達成 "error" の組み合わせに割り当てられた、集合  $M_{00}, M_{01}, M_{10}, M_{11}$  に含まれる学習者数  $N_{00}, N_{01}, N_{10}, N_{11}$  を表 1 の 2x2 分割表に示す。順序理論の依存関係において、テスト項目 X と項目 Y の関係に矛盾するのが、易しいはずのテスト項目 X を間違い、難しいテスト項目 Y を正解している  $M_{01}$  の集合である。テスト項目 X から項目 Y への依存の強さを示す依存定数  $\epsilon$  は式(1)で表され、 $\epsilon$  が小さければ項目間の依存が強いことを意味している。

$$\epsilon = \frac{N_{01}}{n} \quad (1)$$

竹谷らによる認知マップは、人間が地理をどのように把握しているかという観点から、個人における心の中の地図をさしたもので、必ずしも実際の関連や環境を正確に反映したものではない[6]。また、文が有する概念を示す深層構造を生成する目的で、ある事物とそれらの相互間の関連によって概念構造を表現するために意味ネットワーク (semantic net) が提案されている[10]。Schank, R.C. による意味ネットワークを展開させた概念依存関係 (CD : conceptual dependency net) は、言語が表現しようとしている概念を、特定の言語に偏らず純粋に概念に表現しようとしている[7]。このような知識表現においては、ある事物に関連した事物内における概念の構造を示している。

本研究では、学習者の理解順序を知識項目の関連構造として描画することが目的であることから、伊藤らの用いた Airasian & Bart の順序理論を展開させて知識項目間の依存関係を解析する。

### 3. Java 言語の理解状態の測定

#### 3.1 測定のための設定

プログラム言語に対する習熟の差異が及ぼす影響を小さくするため、Java 言語の初心者を対象にし、実験環境を以下のように設定した。

- ① 学習環境の影響を小さくするため、2グループで理解状態を測定し、学習者の総数は 66 名である
- ② 教授者の教育方法の違いによる学習効果の差異

表 1 テスト項目 X/Y におけるの達成 / 非達成 の 2x2 分割表

|       |         | X               |                 | total           |
|-------|---------|-----------------|-----------------|-----------------|
|       |         | error           | correct         |                 |
| Y     | error   | $N_{00}$        | $N_{10}$        | $N_{00}+N_{10}$ |
|       | correct | $N_{01}$        | $N_{11}$        | $N_{01}+N_{11}$ |
| total |         | $N_{00}+N_{01}$ | $N_{10}+N_{11}$ | $n$             |

表 2 テストの正解

Table 2 Parameters of test

|                         |                        |
|-------------------------|------------------------|
| (A1) Variable           | (A14) Interface        |
| (A2) Super class        | (A15) Package          |
| (A3) Class variable     | (B1.1) Class           |
| (A4) Inheritance        | (B1.2) Method variable |
| (A5) Sub class          | (B1.3) new             |
| (A6) Variable           | (B1.4) Instance method |
| (A7) Class              | (B1.5) Instance method |
| (A8) Class              | (B2.1) print           |
| (A9) Instance           | (B2.2) print           |
| (A10) Method            | (B2.3) print           |
| (A11) Instance variable | (B2.4) print           |
| (A12) Super Class       | (B2.5) print           |
| (A13) Method            |                        |

をなくすため、2つの学習者グループに対し 1 人の教授者が講義と演習を行った。

- ③ 問題の難易度の差異を小さくするため、その知識項目に関する基礎的な質問内容にした。

#### 3.2 測定の方法

理解状態の測定は、以下のような手順で行った。

- ① 教授者は、Java 言語の関連の資料から基礎的な知識を学習者に解説する。
- ② 学習者は、例題プログラムの修正とコンパイル、実行といったコンピュータによる演習を行う。
- ③ コンピュータによる演習が終了後、Java 言語に関するペーパーテストを行う。

ペーパーテストは、Java 言語の関連知識が 15 問 (選択式)、プログラムの穴埋めが 5 問 (記述式)、出力結果の回答が 5 問 (記述式) である、テスト項目の正解を表 2 に示す。テストの回答は、項目ごとに正解か間違いのいずれかに判定される。

### 4. 知識項目に関する関連構造の解析

#### 4.1 知識項目の依存関係の解析

知識項目の依存関係の解析において、伊藤らが関

連構造の分析に用いた Airasian & Bart の順序理論の概念を適用する。本研究では、テスト項目の正解を「知識項目」、いくつかの知識項目を集約した1つのグループを「クラスタ」と呼ぶ。

順序理論において、易しいテスト項目 X と難しい項目 Y の関連に矛盾する  $N_{01}$  の全体に対する割合が小さければ、知識項目間の依存関係が成立するとされている。順序理論の式(1)では「まぐれ当たり」や「うっかりミス」が含まれた状態で依存定数  $\epsilon$  が算出されている。式(1)による 600 個の依存定数から 0.1 未満の値を抽出すると 250 個になり、全ての知識項目で依存関係が存在する。この要因として、依存定数  $\epsilon$  に「まぐれ当たり」や「うっかりミス」で生じた回答の影響が考えられる。

順序理論の依存関係において、「まぐれ当たり」や「うっかりミス」が及ぼす依存関係への影響を小さくするため、二項分布の分布関数に偶然の成功と、失敗の確率を適用し、知識項目の依存関係に矛盾率の概念を提案して依存関係を解析した。矛盾率は、2つの回答の組み合わせがある条件のもとで、順序理論の依存関係に矛盾する集合  $M_{01}$  に含まれる確率である。依存関係に矛盾した集合  $M_{01}$  における回答は、易しいテスト項目が間違われ、難しいテスト項目を正解する組み合わせになる。「まぐれ当たり」の確率を“ $f$ ”、「うっかりミス」の確率を“ $c$ ”、テスト項目 Y が「まぐれ当たり」になる回数を“ $k$ ”とする。「まぐれ当たり」の回数が“ $k$ ”であれば、テスト項目 X が「うっかりミス」になる回数は  $N_{01}-k$  で、“ $k$ ”の範囲は  $0 \leq k \leq N_{01}$  になる。

易しいテスト項目 X が “error”，難しいテスト項目 Y が “correct” になる、順序理論の依存関係に矛盾が生じる集合  $M_{01}$  に含まれる学習者を、「矛盾学習者」と呼ぶ。一つの矛盾学習者において、「まぐれ当たり」が確率“ $f$ ”、「うっかりミス」が確率“ $c$ ”で、いずれかの事象が生じていると仮定する。提案手法は以下ようになり、本提案手法による式を式(2)に示す。

$P(N_{00}, N_{01}, N_{11}, f, c)$  は矛盾学習者が集合  $M_{01}$  に含まれる平均確率であり、 $P$  を「矛盾関数」、算出された値を「矛盾率」と呼ぶ。

$B(k; N_{00}+k, f)B(N_{01}-k; N_{11}+N_{01}-k, c)$  は、矛盾学習者が 0 人から  $N_{01}$  人までのそれぞれの学習者数で、集合  $M_{01}$  に含まれる確率を「単独矛盾率」と呼ぶ。

① 二項分布の分布関数  $B$  から、テスト項目 Y における「まぐれ当たり」による偶然の成功の確率  $f$

- で起こる事象が  $(N_{00}+k)$  回のうち、0 人から  $k$  人以下で集合  $M_{01}$  に含まれる累積確率を求める。
- ② 同様に分布関数  $B$  から、テスト項目 X の「うっかりミス」による偶然の失敗の確率  $c$  で起こる事象が  $(N_{11}+N_{01}-k)$  回のうち、0 人から  $(N_{01}-k)$  人以下で集合  $M_{01}$  に含まれる累積確率を求める。
- ③ 矛盾学習者においては、「まぐれ当たり」、「うっかりミス」のいずれかの事象が生じていると仮定する。

図 1.1, 図 1.2 は、集合  $M_{01}$  において  $N_{01}=5$  の時、テスト項目 Y の「まぐれ当たり」が回数  $k$ 、テスト項目 X の「うっかりミス」が回数  $N_{01}-k$  で、集合  $M_{01}$  に含まれる確率を表している。図の目盛りは、縦軸がテスト項目 Y における「まぐれ当たり」が集合  $M_{01}$  に含まれる学習者数、横軸がテスト項目 X における「うっかりミス」が集合  $M_{01}$  に含まれる学習者数を表している。縦軸の目盛り  $y_i$  の長さは、「まぐれ当たり」が確率  $f$  で、 $(N_{00}+k)$  人のうち  $i$  人以下で集合  $M_{01}$  に含まれる確率である。横軸の目盛り  $x_j$  の長さは、「うっかりミス」が確率  $c$  で、 $(N_{11}+N_{01}-k)$  人のうち  $j$  人以下で集合  $M_{01}$  に含まれる確率である。縦軸の“0”から“ $k$ ”までの線分と、横軸の“0”から“ $N_{01}-k$ ”の線分で囲まれた領域は、「まぐれ当たり」が集合  $M_{01}$  に含まれる学習者数が  $k$  である時の単独矛盾率を表している。

図 1.1 は  $k=0$  の事象において横軸の目盛り“0”から“5”までの距離は、

$$B(N_{01}-k; N_{11}+N_{01}-k, c) = \sum_{x=0}^{N_{01}} \binom{N_{11}+N_{01}}{x} c^x (1-c)^{(N_{11}+N_{01})-x},$$

縦軸の“0”の距離が

$$B(k; N_{00}+k, f) = \sum_{x=0}^k \binom{N_{00}}{x} f^x (1-f)^{N_{00}-x}$$

である。前者と後者で囲まれた面積が、この事象における単独矛盾率を表し  $B(0; N_{00}, f)B(N_{01}; N_{11}+N_{01}, c)$  になる。 $k=1$  の事象では、横軸の目盛り“0”から“4”の距離が、

$$B(N_{01}-k; N_{11}+N_{01}-k, c) = \sum_{x=0}^{N_{01}-1} \binom{N_{11}+N_{01}-1}{x} c^x (1-c)^{(N_{11}+N_{01}-1)-x}$$

になり、縦軸の“0”から“1”までの距離は

$$B(k; N_{00}+k, f) = \sum_{x=0}^1 \binom{N_{00}+1}{x} f^x (1-f)^{(N_{00}+1)-x}$$

図 1.2 は  $k=2$  の事象で、集合  $M_{01}$  に含まれる「ま

ぐれ当たり」が2回, 「うっかりミス」が $N_{01}-k$ より3回で, "....." で囲まれた面積が  $B(2; N_{00}+2, f)B(N_{01}-2; N_{11}+N_{01}-2, c)$  になる.

テスト項目 Y の「まぐれ当たり」とテスト項目 X の「うっかりミス」が, 確率  $f$  と  $c$  で集合  $M_{01}$  に含まれている事象において, 上述①と②で囲まれた面積が単独矛盾率を表している.

④  $\sum_{k=0}^{N_{01}} B(k; N_{00}+k, f)B(N_{01}-k; N_{11}+N_{01}-k, c)$  は, 矛

盾学習者が 0 人から  $N_{01}$  人までの単独矛盾率の累積になり,  $N_{01}$  が大きければその累積も大きくなるので,  $(N_{01} + 1)$  で除して平均を求める.

$P(N_{00}, N_{01}, N_{11}, f, c)$  で求められた値は, テスト項目 Y の「まぐれ当たり」が集合  $M_{01}$  に含まれる学習者数  $k$  が, 0 人から  $N_{01}$  人までの単独矛盾率を累積した平均が矛盾率になる.

$$P(N_{00}, N_{01}, N_{11}, f, c) = \frac{1}{N_{01}+1} \sum_{k=0}^{N_{01}} B(k; N_{00}+k, f)B(N_{01}-k; N_{11}+N_{01}-k, c) \quad (2)$$

$$\text{ただし } B(k, n, q) = \sum_{x=0}^k \binom{n}{x} q^x (1-q)^{n-x}$$

$f$ : テスト項目 Y における「まぐれ当たり」の確率  
多岐選択法の場合は選択肢数から偶然に成功する確率を決定した.

$c$ : テスト項目 X における「うっかりミス」の確率  
先行研究[8]の3つの学習フェーズにおける, Java 言語の理解状態の測定データから決定した.

$B$ : 二項分布の分布関数 (確率  $q$  で起こる事象が  $n$  回のうち  $k$  回以下で起こる累積確率)

「まぐれ当たり」の確率  $f$  は, 多岐選択法 (Multiple-choice Method) の問題において, 選択肢が 5 つであり偶然に成功する確率が  $\frac{1}{5}$  であることから  $f=0.2$  とした. 記述式の問題については, 回答として記述される文字の組み合わせが無限に存在するため,  $f=0.0$  とした.  $f=0.0$  であれば,

$$B(k; n, f) = \sum_{x=0}^k \binom{n}{x} f^x (1-f)^{n-x} \text{ が "1" になり,}$$

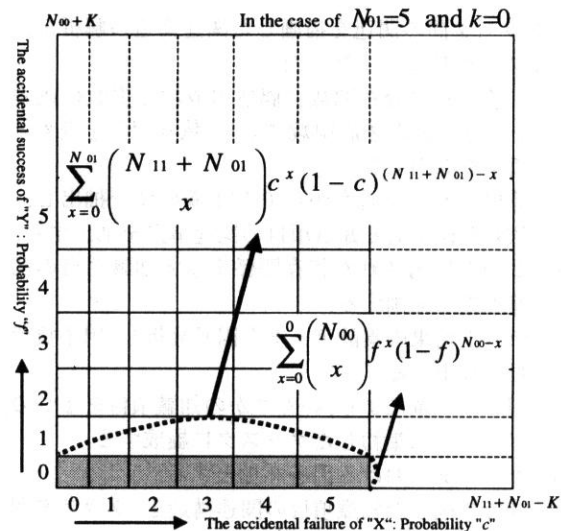
$P(N_{00}, N_{01}, N_{11}, f, c)$  は

$$\frac{1}{N_{01}+1} \sum_{k=0}^{N_{01}} B(N_{01}-k; N_{11}+N_{01}-k, c) \text{ になる.}$$

「うっかりミス」に対しては, 3つの学習フェーズにおける Java 言語の理解状態の測定データ[8]から, 偶然的失敗が生じる確率を  $c=0.247$  とした.

上述のように偶然的成功の確率  $f$  と失敗の確率  $c$  の値を設定し, 矛盾関数  $P$  により矛盾率を算出した.

テスト項目 X と項目 Y の依存関係に矛盾した矛盾



$k$ : The number of times of "the accidental success" of test item "Y"

図 1.1  $k=0$  の時の単独矛盾率

Fig 1.1 Independent inconsistency rate in the case of " $k=0$ "

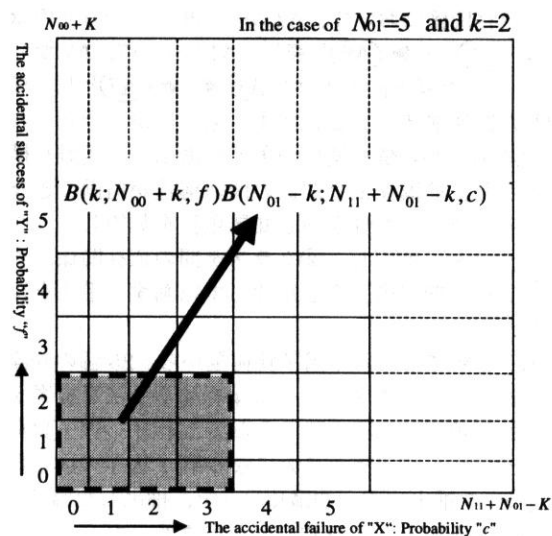


図 1.2  $k=2$  の時の単独矛盾率

Fig 1.2 Independent inconsistency rate in the case of " $k=2$ "

学習者が多ければ、依存関係は小さくなり、逆に矛盾学習者が少なければ依存関係が存在することになる。矛盾学習者となる集合  $M_{01}$  に含まれる確率は、前者が高くなり後者は低くなる。したがって、

$P(N_{00}, N_{01}, N_{11}, f, c)$  が小さければ集合  $M_{01}$  に含まれる確率が低くなり、順序理論の依存関係が存在することになる。

## 4.2 関連構造の解析方法

クラスタ間の関連を描画した関連構造の解析を、以下の手順によって行った。

- ① 二項分布の分布関数に偶然の成功と失敗の確率を取り入れた矛盾関数により、依存関係を表す矛盾率を算出する。
- ② 算出された矛盾率から“0.1”未満の値を抽出し、依存関係にある知識項目の関連構造を抽出する。
- ③ ②から抽出された依存関係を示す知識項目の関連構造を描画する。
- ④ テスト結果の集計データを因子分析し、因子負荷量を算出する。
- ⑤ 因子負荷量により、いくつかの知識項目を1つのグループに集約したクラスタに編成する。
- ⑥ クラスタにおける因子を抽出する。
- ⑦ ③で描画した知識項目の関連構造を、クラスタ間の関連に変換し、関連構造を単純化して描画する。

## 4.3 知識項目からクラスタへの集約方法

式(2)で算出された矛盾率から 0.1 未満の値を抽出した依存関係において、A1 から A9 への一方方向の依存は、 $(A1) \rightarrow (A9)$  と表し、A1 から A7, A7 から A1 といった双方向の依存を  $(A1) \leftrightarrow (A7)$  として描画した関連構造を図 2 に示す。図 2 では、それぞれの知識項目から依存関係の矢印が派生して複雑な構造になり、簡潔な依存関係の関連構造が描画できない。そこで、①特性が同じ知識項目を集約して一つのクラスタに編成し、②クラスタ間の依存関係で表し、③その関連構造を単純化して描画する手法を提案した。

因子分析で得られる因子負荷量から、25 個の知識項目をいくつかのクラスタに集約する手法を提案する。知識項目  $j$  の第一因子から第  $n$  因子の因子負荷量において、絶対値が最大になる第  $i$  因子の因子負荷量  $a_{ij}$  を抽出する。知識項目  $j$  を、抽出した第  $i$  因子のグループとして分類し、このグループをクラスタ  $f(i)$  として編成した。

表 3 因子負荷量による知識項目の分類

Table 3 Classification of the knowledge items by factor loadings

| factor                      | items               | factor                | items                   |
|-----------------------------|---------------------|-----------------------|-------------------------|
| f(1) Method, Code           | (B2.4) print        | f(6) Thought          | (A2) Super class        |
|                             | (B2.5) print        |                       | (A15) Package           |
|                             | (B2.1) print        |                       | (A14) Interface         |
| f(2) Class, Variable        | (A1) Variable       | f(7) Class, Interface | (A8) Class              |
|                             | (A7) Class          |                       | (B1.5) Instance method  |
| f(3) Variable, Code         | (A6) Variable       | f(8) Class, Instance  | (B1.2) Method variable  |
|                             | (B2.2) print        |                       | (A9) Instance           |
| f(4) Super Sub Class, Class | (B2.3) print        | f(9) Variable, Method | (B1.1) Class            |
|                             | (A5) Sub class      |                       | (B1.4) Instance method  |
|                             | (A12) Super Class   |                       | (A11) Instance variable |
| f(5) Class, Variable        | (A10) Method        |                       | (B1.3) new              |
|                             | (A3) Class variable |                       | (A4) Inheritance        |
|                             |                     |                       | (A13) Method            |

本分析では 9 因子による因子分析の因子負荷量を

もとに、25 個の知識項目を 9 つのクラスタに編成した。

それぞれのクラスタの因子を次のように抽出した。ここで、クラスタ f(2), f(4), f(7), f(8), f(9) においては、一つ概念では説明できないため、関連ある 2 つの概念で表した。また、クラスタ f(6) を構成する知識項目 A2 は、問題を熟読して常識的知識をもとに、消去法により回答群の中から見つけ出すことができる。したがって、この知識項目に関する問題が、思考力に影響を受けることから、この因子を「Thought: 思考」として抽出した。因子分析から抽出したクラスタの因子と、クラスタに含まれる知識項目を表 3 に示す。

## 4.4 クラスタの関連構造の単純化

知識項目の依存関係を、上述の提案手法によって編成したクラスタの関連構造として描画した。クラスタ間の依存関係を描画した関連構造を図 3.1 に示し、関連構造を単純化する方法を以下に示す。

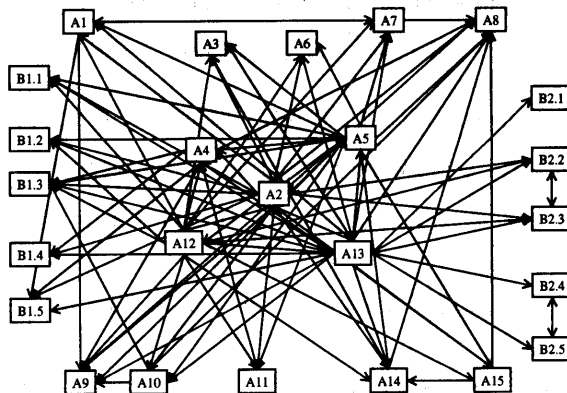


図 2 知識項目の関連構造

Fig 2 The relational structure of the knowledge items

図 3.1においてクラスタ f(6)の知識項目 A2(以下、本節ではそれぞれのクラスタと知識項目を f(6), A2 というように呼ぶ。) から f(3) の B2.2 と B2.3 へ、依存関係を表す 2 本の矢印 "→" で描かれているが、これらを f(6)から f(3) への依存関係とし、1 本の矢印 "→" で描画する。同様に双方向の依存関係は "↔" で表し、これらを図 3.2 に示す。

図 3.2 の f(6)から f(3)の依存関係は、 $f(6) \rightarrow f(3)$  と  $f(6) \rightarrow f(9) \rightarrow f(3)$  の依存関係が並行して存在している。後者の依存関係は、f(6)を理解して f(9)が理解され、次に f(3)の理解が成立しており、f(9)の理解を介して f(3)への依存関係が継承されている。このような依存関係では、後者の依存関係が成立すれば前者の依存関係も含まれることになり、

$f(6) \rightarrow f(3)$  の依存関係は  $f(6) \rightarrow f(9) \rightarrow f(3)$  に包括して表した。他に並行する依存関係が存在する f(6)から f(7), f(4)から f(7)などの依存関係についても同様の処理を行った。

f(6)から f(4), f(6)から f(9), f(4)から f(9)は、双方向の依存関係が存在するが、矛盾率の平均値が小さい方向を依存関係が強いとして一方向の矢印に変換した。これらのクラスタの依存関係を  $f(6) \rightarrow f(4)$ ,  $f(6) \rightarrow f(9)$ ,  $f(4) \rightarrow f(9)$  として表した。この手法により、依存関係の関連構造を単純化して描画した。

上述の処理をした依存関係を、" $\rightarrow$ " だけでクラスタ間の依存関係の関連構造を単純化して描画すれば、図 3.3 のようになる。

## 5. クラスタの関連構造の解析

クラスタ f(6)の因子が "Thought" であることから、クラスタの関連構造の起点でなっていることが説明できる。クラスタ f(4)の Super Class, Sub Class に関する知識項目が、実際に Java 言語を理解する最初の概念になっている。クラスタ f(4)に関連した Class に関する理解から、クラスタ f(9)の Variable, Method の理解へと発展している。

クラスタ f(9)は、プログラム Code と、Class を含む 2 つのクラスタのグループに関連している。この関連構造から、クラスタ f(9)の Variable と Method に関する知識項目が、Java 言語の学習の中心になっていると考えられる。

またクラスタ f(9)は、4 つのクラスタに関連している。したがって、この因子である Variable, Method を理解すれば、他のクラスタにおける知識項目の理

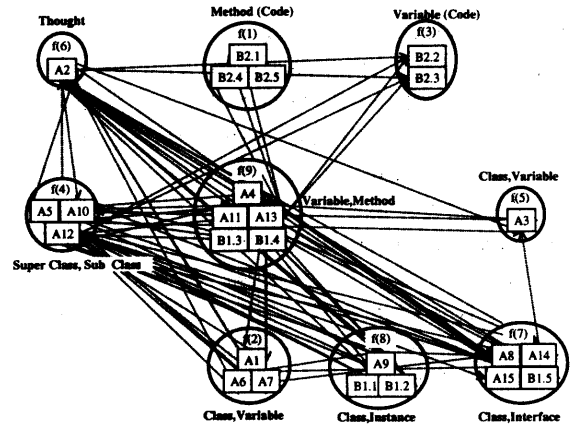


図 3.1 クラスタに集約された知識項目の関連構造  
Fig 3.1 The relational structure of the collected knowledge items in the cluster unit

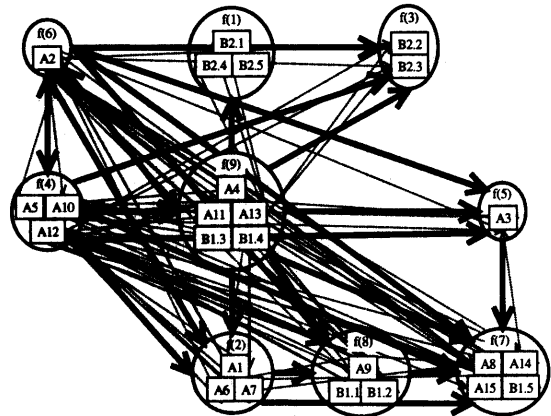


図 3.2 クラスタの関連構造

Fig 3.2 The relational structure of the clusters

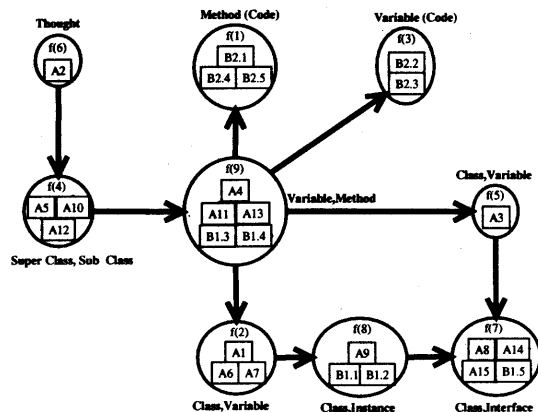


図 3.3 単純化されたクラスタの関連構造

Fig 3.3 The relational structure of the simplified clusters

解が促進されると考えられる。

図 3.3 クラスタの関連構造から、Java 言語の知識項目における理解順序について次のように考察した。

- ① 学習の最初は、Class の概念の理解である。
- ② Variable, Method の概念が、Java 言語の学習と理解の中心である。
- ③ Variable, Method の概念の理解が進むと、Code と Class の概念に関する理解へと展開していく。
- ④ Variable, Method の次の学習は、Code と Class の概念に関する知識項目に分けて進めることが可能である。
- ⑤ クラスタ f(7)の Class, Interface に関する知識項目の学習は、クラスタ f(5), f(8)から関連を受けているため、複合的な概念の理解が必要になる。
- ⑥ Code と Class の概念に関するクラスタの理解順序は、この関連構造図からは判別できない。

## 6. まとめ

本研究において、Java 言語における知識項目の依存関係の関連構造を描画し、理解順序を解明できた。25 個の知識項目の複雑な関連構造を、9 つのクラスタに編成し、クラスタ間の関連構造として簡潔に描画できた。順序理論の依存関係において、「まぐれ当たり」と「うっかりミス」が及ぼす依存関係への影響を小さくするため、二項分布の分布関数に偶然の成功と失敗の確率を適用し、知識項目の依存関係に矛盾率の概念を提案して知識項目の依存関係を解析した。Java 言語の複雑な知識項目の関連を単純化して描画するため、因子負荷量から知識項目をいくつかのクラスタに編成する方法と、クラスタの関連構造を単純化する方法を提案した。

Java 言語の学習に関する実験から、以下のような結果と知見が得られた。

- ① 提案した矛盾関数によって、600 個の矛盾率を算出できた。
- ② 因子負荷量から、25 個の知識項目を 9 つのクラスタに集約し、クラスタの因子を抽出できた。
- ③ 600 個の矛盾率から“0.1”未満の数値を抽出し、依存関係が存在する知識項目の組み合わせを 104 個抽出した。
- ④ ③の組み合わせを、知識項目を集約したクラスタ間の関連に変換し、その関連構造を単純化して描画した。

知識項目を集約したクラスタの関連構造は、Java 言語の教育あるいは学習の順序をクラスタの単位で決定することができる。また、クラスタに含まれる

知識項目は、同じ特性の因子であることから、一つのクラスタに含まれる知識項目を一つの単元とした教育や学習の体系化が考えられる。知的教育システムにおける学習者モデルの構築では、知識項目を集約したクラスタの関連構造は有効な知見であると考えられる。今後は、本関連構造を利用した知的教育システムの開発へ展開したいと考えている。

## 参考文献

- [1] P.W.Airasian, W.M.Bart, “Ordering Theory: A New and Useful Measurement Model,” *Education Technology*, pp.56-60, May, 1973.
- [2] 長谷川聡, 山住富也, 小池慎一, “プログラミング教育における制御構造のイメージと理解度について,” *情報処理学会論文誌*, Vol.39, No.4, pp.1180-1183, 1998.
- [3] 伊藤公紀, 大内東, “ファジィ項目関連構造分析による学習者の特性解析,” *情報処理学会誌, コンピュータと教育* 24-4, pp.27-34, 1992.
- [4] 岩根典之, 竹内章, 大槻説乎, “算数の文章題を対象としたネットワーク型知的教育支援環境,” *電子情報通信学会論文誌*, Vol.J80-D-II, No.4, pp.915-924, 1997.
- [5] 松田憲幸, 柏原昭博, 平嶋 宗, 豊田順一, “プログラムの振舞いに基づく再帰プログラミングの教育支援,” *電子情報通信学会論文誌*, Vol.J80-D-II, No.1, pp.326-335, 1997.
- [6] 西田豊明, 荒木雅彦, “空間的知識の表現と利用,” *認知科学ハンドブック*, 第VI編, 第3章, 共立出版, pp.293-294, 1995.
- [7] R.C.Schank, “*Conceptual Information Processing*,” New York: North-Holland, 1975.
- [8] Y. Takemura, K. Shima, K. Matsumoto, K. Inoue, and K. Torii, “Factor Analysis of Comprehension State in the Learning Phases of a Programming Language,” *Proc. of the 6th Asia Pacific Software Engineering Conf. [APSEC'99]*, pp.136-143, Takamatsu, Japan, Dec. 1999.
- [9] 竹谷 誠, 佐々木 整, “学習者描画の認知マップによる理解度評価,” *電子情報通信学会論文誌*, Vol.J80-D-II, No.1, pp.336-347, 1997.
- [10] 田村進一, 柳原圭雄, 唐沢博, “人工知能の世界,” *技術評論社*, pp.169-170, 1992.
- [11] 寺野隆雄, “常識とコンピュータ・システム,” *認知科学ハンドブック*, 第VI編, 第2章, 共立出版, pp.284-285, 1995.