事例に基づく初等アセンブラプログラミング評価支援システム

渡辺博芳† 荒井正之† 武井惠雄†

本論文では初等アセンブラプログラミング授業において，提示した課題に対して学生が作成したプログラムの評価作業を支援する手法を提案し，提案手法に基づいて実現した，CASL を教材とした授業におけるプログラム評価支援システムについて述べる．本研究で対象とする評価作業は，評価対象のプログラムが提示した問題の題意を満たしているかどうかの判定とそのプログラムに対するアドバイスの作成である．本手法では，そのような評価作業を計算機で行うために，(1) プログラムの動作の評価と，(2) 過去の評価事例に基づくプログラムの実現方法の評価を行う．また，実現した評価支援システムにおいては，事例の豊富さに応じて評価モードを選択可能，教員を支援するためのウェブベースの使いやすいインタフェース等の特徴を持たせた．本システムを実際の授業で用いて評価を行ったところ，教員のプログラム評価の作業負荷を大幅に減少できることが示された．

Case-based Evaluation Support System of Novice Programs Written in Assembly Language

Hiroyoshi Watanabe,† Masayuki Arai† and Shigeo Takei†

In this paper, we propose a method of evaluating students' programs written in assembly language. The target evaluation tasks are to judge whether a student's program satisfies the requirements of the given problem and to give advice for the student program. The proposed method to perform the evaluation tasks consists of two processes: (1) evaluating the program's action and (2) evaluating the implementation based on cases. We implemented a support system of evaluating CASL programs based on the method. The implemented system has a web-based user interface to support teachers' evaluation work. The system was utilized for actual classes and the results showed that the system reduced the teacher's evaluation work drastically.

1. まえがき

プログラミング教育においてプログラミング演習の果たす役割は大きい．プログラミング演習授業は作品制作のような形態をとることもあるが，特に初等プログラミング教育においては，教員が提示した問題の題意を満たすようなプログラムを学生が作成し，提出されたプログラムやレポートを教員が評価するという形態をとることが多い．しかし，学生数が多くなると，教員の評価作業は増大する．このようなプログラムやレポートの評価において，総合的な評価は教員が行うべきであるが，比較的低レベルな作業を計算機に行わせることで，評価作業を効率化できるはずである．そこで，本研究では，提示した課題に対して学生が作成したプログラムが題意を満たしているかどうかの判定と学生に対するアドバイスの作成を支援するシステムを実現することを目指す[1)~3)]．

プログラミングの教育支援，あるいは学習支援に関する研究はさかんに行われており，トータルな演習環境を実現しようとするもの[4)~6)]から，教材の実装[7)]，プログラムの振舞いに基づく理解の支援[8)]，プログラム理解に基づく誤りの指摘[9)~16)]，提出されたプログラムの評価やレポート処理に対する支援[17)~19)]等様々なアプローチがある．これらのうち，本研究はプログラミングの演習授業におけるプログラム評価を支援するアプローチに分類される．

本研究で実現しようとする評価支援システムの概略を**図 1** に示す．学生は電子的にプログラムを提出し，そのプログラムに対する判定結果やアドバイスを受け取る．評価支援システムは学生が提出したプログラムに対する分析を行い，判定結果やアドバイスを生成する機能を持つ．教員はサーバにアクセスすることで，提出状況や提出されたプログラム，システムの分析結果等を閲覧し，判定やアドバイスを書き込むことができる．学生が提出したプログラムに対して，システム

† 帝京大学理工学部
School of Science and Engineering, Teikyo University

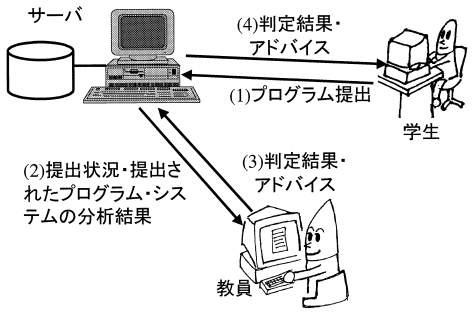

図 1 評価支援システムの概要
Fig. 1 An outline of an evaluation support system.

の分析結果をもとに最後は教員が判定を行うことを原則とするが，システムが判定可能な場合には自動的に判定結果を送付することも可能なシステムにする．システムが自動的に判定するか，教員が判定を行うかは，提示する問題ごとに設定できるものとする．

本論文では，CASL を用いた初等アセンブラプログラミングを対象として，図 1 のような評価支援システムを実現する．その際に，(a) 学生のプログラムに対する判定結果やアドバイスを生成する機能をどのように実現するか，(b) 評価支援システム全体としてどのように実現するかという 2 つの問題がある．本論文では，(a) について事例ベース推論[20),21)]を応用した評価法を提案する．また，(b) については，状況に応じて 3 つの評価モードを設定可能とし，教員が提出状況閲覧やプログラムの評価・アドバイスを行うための使いやすいインタフェースを実現する．

以降，2 章でプログラム評価の方針とシステムのプログラム評価処理の全体的な流れについて述べ，3 章ではプログラムの動作の評価方法，4 章で事例に基づくプログラムの評価法について述べる．ここまでが学生のプログラムに対する判定結果やアドバイスを生成する機能を実現する方法論である．5 章では実際に実現したプログラム評価支援システムの全体について述べる．6 章以降で，実際の授業における実験結果と考察について述べ，まとめる．

2. 評価方針と評価処理の流れ

2.1 評 価 基 準

教員が問題を提示する際には，学生に習得させたい概念等に関する教育的な意図がある．教員は提出されたプログラムやレポートを分析し，意図していた概念を学生が習得したかどうかの評価を行い，学生にアドバイスをしたり，場合によってはプログラムやレポートの再提出を求める．本研究が対象とする評価作業は，このような，学生が作成したプログラムが提示した問題の題意を満たしているかどうかを判定する作業である．以降では，題意を満たしている場合は「合格」，そうでないときは「不合格」といい，この題意を満たすかどうかの評価作業を「合否判定」と呼ぶことにする．「不合格」という表現は，現在判定対象となっている，そのプログラムが不合格であることを示し，提示された問題について学生が不合格となるのとは異なることに注意されたい．

合否判定は以下の 2 つの基準に基づいて行う．

(a) プログラムの動作が問題の題意を満たしていること．

(b) プログラムの実現方法が問題の題意を満たしていること．

(a) は問題を解くという意味で必須条件である．(b) は上で述べた教育的な意図が達成されたかどうかということに関連する．たとえば，出題された問題文に「スタックを用いて」という記述が含まれているとき，教員は，学生がスタックについて学ぶことを意図している．この場合，題意どおりに動作するプログラムであっても，スタックが使われていなければ，教員は「実現方法は題意を満たしていない」と判断するであろう．

2.2 評 価 方 針

前節で述べた評価基準をテストするために，プログラム評価処理は，プログラムの動作の評価とプログラムの実現方法の評価の 2 つのフェーズに分けて行うこととする．

プログラムの動作評価はあらかじめ複数のテストデータを用意しておき，それらのデータに対する動作をテストする．このような評価は，人間よりもむしろ計算機の方が得意であり，完全に自動化可能である．

プログラムの実現方法の評価は，評価事例を利用するアプローチ，すなわち，事例ベース推論[20),21)]のアプローチをとる．事例ベース推論は，与えられた問題に類似する過去の問題解決事例を直接利用して問題解決を行うような推論法である．事例に基づくプログラム評価は，「あるプログラムの評価を行う際に，過去の事例を検索し，評価対象のプログラムと同じ実現方法と見なせるプログラムの評価事例が存在すれば，その事例の判定結果とアドバイスを評価対象のプログラムに適用すること」である．

事例ベース推論のアプローチをとる利点は 2 つある．1 つはプログラムの評価に必要な知識やヒューリスティクスが少ないことである．プログラムの実現方法の評価を，従来の知識に基づくプログラム理解やプログラム認識のアプローチ[9)~14)]で行う場合，非常に

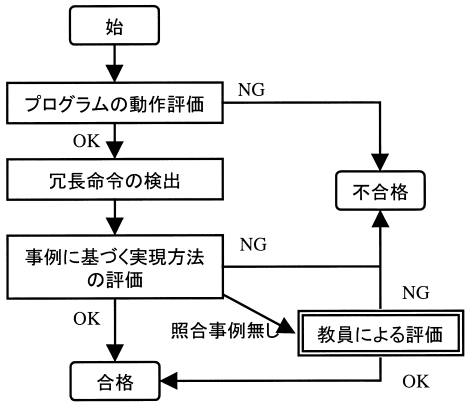

図 2 プログラム評価処理の流れ
Fig. 2 Flowchart for the evaluation of students' programs.

多くの知識・ヒューリスティクスが必要であり，現実的でない．2 つめの利点は，事例の追加によってシステムの能力が向上できることである．教員がプログラムの評価を行うたびに事例を追加するようにしておけば，それ以降，それと同様なプログラムは事例に基づいて評価を行うことができる．つまり，システムを使うことによって，システムの能力（評価可能なプログラムの数）を高めることができる．また，教員が前もって用意した解答例も事例と見なすことができるので，ある時点では，教員が前もって用意した解答例と，それまでに獲得した評価事例が利用可能である．

2.3 評価処理全体の流れ

プログラム評価処理の流れを**図 2** に示す．最初に，学生に質問して得たラベル情報を利用して，プログラムの動作を評価する．ここで，アセンブル可能性とテストデータに対する動作をテストし，アセンブルエラーがある場合や正しく動作していないと判定された場合は，システムの出力は不合格となる．動作評価の詳細は，3 章で述べる．

正しく動作すると判定された場合は，次に冗長命令の検出処理を行う．冗長命令とは，その命令がなくともプログラムの動作に影響を与えないような命令（削除可能な命令）である．冗長命令の検出処理は合否判定には直接関係なく，冗長命令情報は，事例照合処理において用いられる．また，教員が判定を行ったり，アドバイスを記述したりする際に参考として使用する．冗長命令の検出は，プログラムリストの命令を 1 つずつ削除してはプログラムの動作を確認することによって行う．

最後に，事例に基づく実現方法の評価を行う．ここで適用可能な事例が存在すれば，その事例の判定結果を適用する．すなわち，事例の判定結果が合格であればシステムの出力も合格となり，事例の判定結果が不合格であればシステムの出力も不合格となる．また，事例のアドバイスに簡単な修正を施し，アドバイスを生成する．適用可能な事例が存在しない場合については，システムは判定とアドバイスを生成できない．実現方法評価の詳細は，4 章で述べる．

```
問題識別子： 0001
問題名： 問題 1
キーワード： 指標レジスタ，ループ，合計
問題文：
メモリの連続した領域に N 個の整数が格納されている．
指標レジスタを用いて，これらの合計を求め，結果を
メモリに格納するプログラムを作成せよ．なお，デー
タの個数 N はメモリに格納されているものとする．
ラベル情報：
N       データ数のある場所のラベル
DATA    データの先頭のラベル
SUM     データの合計（答え）の場所のラベル
テストデータ：
        入力：
        N       1       3
        DATA    3       -5  3   0
        出力：
        SUM     1       -2
```

図 3 問題情報の例
Fig. 3 An example of the problem information.

3. プログラムの動作の評価

3.1 問題情報の表現

1 つの問題についての問題情報を 1 つのテキストファイルに記述する．問題情報の例を**図 3** に示す．問題情報は以下の情報からなる．

- 問題識別子（全体でユニークな識別子）．
- 問題名（授業等で出題の際，参照するための名称）．
- キーワード．
- 問題文．
- ラベル情報（テストデータや事例で用いられるラベルの情報）．
- 動作評価のためのテストデータ．

図 3 では，テストデータは 1 組しか示していないが，実際には，このような入力と出力の組を複数組用意する．1 組のテストデータは，複数の入力データと複数の出力データの組である．入力データ，出力データとも，ラベル名，メモリサイズ，値の列で表現する．値の列は CASL で許されている 10 進定数，16 進定数，文字定数のいずれかである．図 3 のテストデータの例では，ラベル N で示される記憶領域のサイズが 1

で，そこに 3 が設定されており，ラベル DATA から 3 語に −5，3，0 というデータが定義された状態でプログラムを実行すると，ラベル SUM の部分に −2 が求められることを示している．

3.2 プログラム動作評価処理

動作評価処理では，ラベルの対応情報の取得を行った後，テストデータの組数だけ，データ部の書き換え，アセンブル・実行，出力ラベルの検査を行い，動作を確認する．

3.2.1 ラベルの対応情報の取得

ラベルの対応情報の取得では，問題情報におけるラベル名と学生のプログラムで用いられているラベル名の対応をとるために学生に質問する．質問は図 3 で示すラベル情報を用いて，たとえば，「君のプログラムでデータ数のある場所のラベルは何ですか？」のような質問をする．この質問に学生が KOSU と答えれば，問題情報の N というラベルは学生のプログラムでは KOSU という名称であることが分かる．このようなラベルの対応付けはプログラム認識を行うことで推定は可能であるが，学生が自分のプログラムに関する質問に答えることも教育的であるので，このようなアプローチをとる．また，質問をするのではなく，システムが問題文に示すラベル名をデフォルトとして提示し，学生が必要に応じて修正することも可能である．

3.2.2 プログラム動作の確認処理

以下の処理をテストデータの数だけ行う．

(1) 取得したラベルの対応情報とテストデータを基に学生のプログラムのデータ部分を書き換える．
(2) データ部を書き換えたプログラムをアセンブルし，レジスタの初期値と DS 命令で確保されるメモリ領域の初期値として 0 以外のランダムな値を設定して実行する．
(3) 出力に定義されたラベルについてテストデータと同じ値になっているかを比較する．

正しく動作しないテストデータが 1 組でもあれば，不合格である．すべてのテストデータについて正しく動作すれば，次の処理に進む．正しく動作しない場合は，レジスタの初期値と DS 命令で確保されるメモリ領域の初期値を 0 に設定して同様に動作の確認を行う．これで正しく動作した場合は，レジスタやメモリ領域の初期化について助言を与える．

4. 事例に基づくプログラム実現方法の評価

4.1 事例の表現と一般化ルール

4.1.1 事例の表現

事例ベース推論における事例は，一般に問題記述と解記述，あるいは解法から構成される．本手法において，事例は，プログラムリスト，判定結果（合格，または不合格），アドバイス，事例作成年月日，事例作成者から構成する．これらのうち，プログラムリストが問題記述，判定結果とアドバイスが解記述である．事例の例を**図 4** に示す．

```
PRG     START
SET     GRX1 0
        SET     GRG2 0
LOOP    ADD     GRG2,DATA,GRX1
        INC     GRX1 1
        COMP    GRX1 N
        J+-OR-  LOOP
        ST      GRG2,SUM
        EXIT
SUM     DS      1
N       DC      3
DATA    DC      1
        DC      2
        DC      3
        END
事例識別子： c001@000001
判定： accept
アドバイス：
作成年月日： 1999 6 23
作成者： 渡辺博芳
```

図 4 事例の例
Fig. 4 An example of a case.

プログラムリストは CASL プログラム，または CASL プログラムの一般化表現である．一般化表現を用いる目的は，1 つのプログラムリストが複数のバリエーションと照合できるようにすることである．

CASL の一般化表現は，汎用レジスタの表現と一般化表現として新しく導入された命令以外は CASL の文法に準拠する．汎用レジスタについては，同じ役割を持つレジスタに対して同じ値を割り当てる．この値はレジスタ番号ではないので，CASL の汎用レジスタは 0 から 4 であるが，4 を超える値も用いられる．また，一般化表現として新しく導入する命令は，同じ処理を表す複数通りの CASL の記述に照合する表現である．一般化表現の命令と通常の CASL の文法に従うコードは一般化ルールと呼ぶプロダクションルールによって対応付ける．

4.1.2 一般化ルール

図 4 で，SET，INC，COMP，J+-ORJ-は一般化表現として新しく導入した命令であるが，これらのうち，SET に関する一般化ルールの例を**図 5** に示す．なお，本システムで用いたすべての一般化表現を付録に示す．図 5 において，? で始まる記号は変数を表して

```
SET   RG1   ?C1   ⇔          LEA   ?RG1   ?C1

SET   RG1   ?C1   ⇔          LD    ?RG1, ?L1
                       ?L1   DC    ?C1
```

図 5 一般化ルールの例
Fig. 5 Examples of generalization rules.

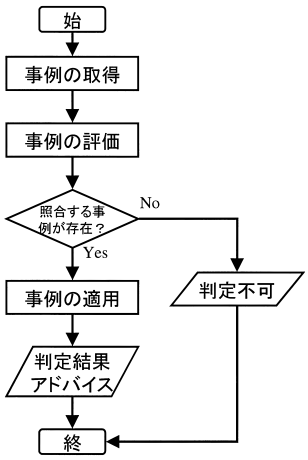

図 6 事例に基づくプログラム評価処理の流れ
Fig. 6 Flowchart for the case-based program evaluation.

いる．汎用レジスタを表す変数は ?RGn，指標レジスタを表す変数は ?RXn，ラベルを表す変数は ?Ln，定数を表す変数は ?Cn のような形式をとり，n は 1 以上の整数である．SET 命令は，汎用レジスタに値を設定する命令であるが，汎用レジスタに値を設定するには，LEA 命令を使う方法と LD 命令で別の領域に DC 命令によって定義された値を設定する方法がある．図 5 の上が前者，下が後者である．

4.2 事例に基づくプログラム評価処理

一般に，事例ベース推論処理は，事例の検索および評価による事例の選択，選択した事例の修正および適用のプロセスからなる．本手法における事例に基づくプログラム評価処理の流れを**図 6** に示す．最初に，評価対象のプログラムと同じ問題に解答したプログラムに関するすべての評価事例を取得する．次に，取得した個々の事例のプログラムリストと評価対象のプログラムの照合処理を行い，最も照合度の高い事例を選択する（事例の評価）．照合する事例が存在すれば，その事例を適用し，判定結果とアドバイスを出力する．照合する事例が存在しない場合は判定不可となる．以下に，事例の評価処理，事例の適用処理について詳述する．

4.2.1 事例評価の方針

事例の評価では，事例と評価対象の照合処理を行い，評価対象が事例と照合するか否か，照合する場合は照合度はどれくらいかを判定する．

(1) アセンブラプログラムの照合処理

アセンブラプログラムは，個々の行が 1 つの命令を表しているので，2 つのプログラムリストにおいて，命令の対応をとることができる．命令の対応関係は，命令コードが等しく，オペランドがそれぞれのプログラムで同じ役割のレジスタやラベルであるという基準で決めることができる．また，4.1 節で述べた一般化表現の命令については，一般化ルールによって導かれる CASL コードに対して同様の基準を適用する．このような基準に基づいて，事例のプログラムリストと評価対象のプログラムリストの間で，命令，ラベル，レジスタの矛盾のない対応をとる処理が照合処理である．

(2) 照合条件

以下のような条件を満たす場合,「事例と評価対象は照合する」という．

- 事例のプログラムのすべての命令が評価対象プログラムの命令に対応先を持ち，事例のプログラムの命令に対応付けられない評価対象プログラムの命令が冗長命令である．

このような場合，事例と評価対象プログラムは基本的に同じ実現方法であると見ることができる．

(3) 完全照合

以下のような条件を満たす場合を「完全照合」と呼ぶ．

- 事例と評価対象が照合するケースで，特に，個々の命令が 1 対 1 に対応しており，それらの順序関係も等しい．

完全照合ならば，2 つのプログラムはまったく同じと考えられるので，事例の判定結果とアドバイスが評価対象にそのまま適用できる．

(4) 照合度

照合度を判定するうえで，最も重要なのは完全照合かどうかである．完全照合の場合は，システムの評価結果を教員がチェックせずに直接学生に通知することも可能である．照合するが，完全照合でない場合は教員の最終的な評価を必要とする．

照合度情報は，照合する複数の事例の中から適用する事例を選択するためにも用いる．本システムでは照合度を値として求めないが，以下のような基準で照合度の高低を比較することとした．すなわち，完全照合でないケースには，(a) 対応する命令対の順序関係が異なるケース，(b) 評価対象プログラムに事例のプログラムに含まれない冗長命令があるケース，(c) その両方のケースがある．本システムでは，(b) よりも (a) を優先することとし，(1) 対応する命令対の順序関係

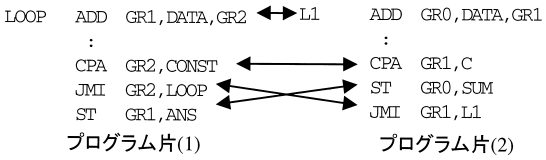

図 7 対応する命令の順序関係の違い
Fig. 7 Differences of the order of corresponding instructions.

の異なる部分の数をカウントし，その数が最も少ない事例を選択する．(2) 順序の異なる部分の数が同じであれば事例に対応先を持たない命令の数が最も少ない事例を選択する．

(5) **対応する命令対の順序関係**

図 7 に示す 2 つのプログラム片における命令の対応関係で，ST 命令の対応と JMI 命令の対応は順序関係が異なっている例である．対応する命令対の順序関係が異なると，判定やアドバイスが異なる可能性がある．たとえば，図 7 において，ST 命令をループの中におくか，外におくかの違いによってアドバイスが異なると考えられる．ただし，順序関係の違いが重要でない部分を，一般化表現プログラムで指定できる．そのような部分についての順序の違いは，照合度を判定する際に，順序の違いとは見なさない．

4.2.2 事例の評価処理

事例評価処理の詳細を**図 8** に示す．照合処理は，命令，ラベル，レジスタの対応関係の候補を生成した後，対応関係の候補を他の 2 つの対応関係の情報を用いて絞り込むことで行う．たとえば，ラベルの対応関係とレジスタの対応関係の情報を用いて，命令の対応関係の候補を絞り込むことができる．他の対応関係についても同様である．このようにして，命令，ラベル，レジスタの対応関係に矛盾がなく，前項で述べた照合条件を満たせば，事例と評価対象のプログラムは「照合する」と判定され，照合処理は成功する．

照合が成功した場合は，事例と評価対象間で対応する命令の順序関係の違いを評価し，順序関係が異なる部分を切り出す．それまでの照合処理で最も照合する事例と今回の事例の照合度を比較する．より照合度が高いと判断される場合は，最も照合する事例を置き換える．事例ではプログラムリストを一般化表現で記述している場合があるが，照合度が同程度の場合は，より特殊な（一般化の度合いが低い）方を優先する．一般化の度合いは，一般化表現の命令数で表す．すなわち，一般化表現の命令数の多い方が一般化の度合いが高いとして評価を行う．それも同じ場合は，新しい事例を優先する．

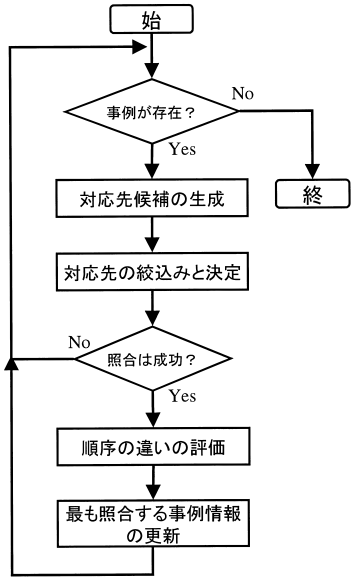

図 8 事例の評価処理の流れ
Fig. 8 Flowchart for the case evaluation.

4.2.3 事例の適用

評価対象に照合する事例が存在する場合は，その事例の判定結果を評価対象プログラムの判定結果とする．また，事例のアドバイス文を以下のように修正して評価対象プログラムに対するアドバイスとする．

- 第 x 行，x 行目という表現の x という数字を事例のプログラムリストの x 行目の命令に対応する評価対象プログラムの命令の行に修正する．
- 事例のプログラムのレジスタ名，ラベル名を表す文字列はそれに対応する評価対象プログラムのレジスタ名，ラベル名に置き換える．

システムは，以上のようにして求めた判定結果，アドバイスのほかに，適用した事例の識別子，その事例との照合における順序関係の違い情報，冗長命令情報を出力する．これらのうち，学生に通知されるのは判定結果とアドバイスのみであり，その他の情報は教員が最終的な評価を行う際に参照する．冗長命令については，検出した冗長命令のうちから，教員がアドバイスが必要と判断したものについてのみ，アドバイスを行う．したがって，冗長命令に関しても事例のアドバイス文に含まれるもののみが再利用される．

5. 実現したシステム

5.1 システムの概要

図 9 は実現したシステムの概要を学生に説明するために用いる図である．つまり，図 9 は表向きの（学生の側から見た）システム像を表している．この図を

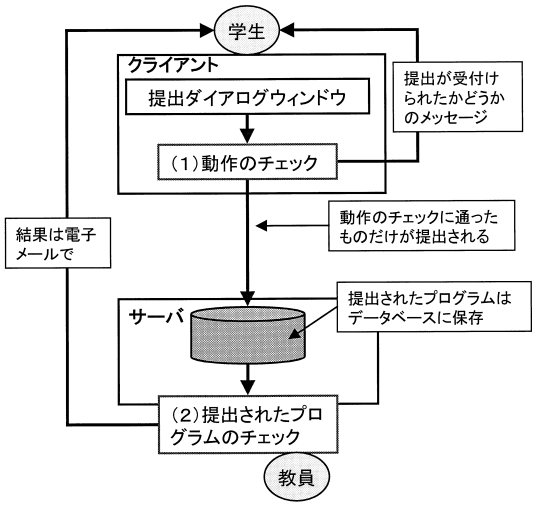

図 9 実現したシステムの概要
Fig. 9 An outline of the implemented system.

用いて，実現したシステムにおいてプログラムが評価される過程を説明する．

(1) 学生は，学生用クライアントシステムの提出ダイアログウィンドウからプログラムを提出する．提出する際に，学生は 3.2.1 項で述べたラベル情報に関する質問に回答する．

(2) 学生用クライアントシステムは提出されたプログラムの動作の評価を行う．正しく動作していると判定されたプログラムだけが「提出受付」となる．ここで提出が受け付けられたか，動作が正しくないかによって，その旨のメッセージを提示する．

(3) 提出されたプログラムはサーバに保存され，サーバにおいて実現方法の評価を行う．評価結果に基づいて，判定結果（合格，または再提出）とアドバイスを電子メールで送付する．

5.2 評価モード

実際の授業の多様なニーズに応えるために，このシステムの判定結果の利用形態として，3 つのモードを定義した．以下にそれら 3 モードについて説明する．

(1) **自動モード**

プログラムが提出された時点で，事例との照合を行い，事例と完全に照合した場合，システムの判定結果とアドバイスを自動的に学生に通知するモードである．事例と完全に照合しない場合のみ，教員が判定作業を行うので，教員の作業負荷の軽減の効果が大きい．一方で，システムにより自動的に判定されるプログラムと教員が判定するプログラムでは，判定結果の通知の時間差が大きいという欠点がある．つまり，ある特定のプログラムについてのみ，時間的に早く合格が通知された場合に，早く合格通知が届くプログラムの方が良いプログラムであるという誤解を学生に与える懸念がある．したがって，自動モードは，簡単な問題や穴埋め問題等，解答のバリエーションが少ない場合や，事例が豊富にある場合に用いると効果的である．

(2) **手動モード** システムの判定結果を基に，最後は必ず，教員が判定を行うモードである．提出されたプログラムに照合する事例が存在する場合は，照合した事例に対する判定結果を参照できることで，教員の作業負荷は軽減される．ただし，教員は提出を受け付けたすべてのプログラムを処理をするので，作業負荷軽減の効果は自動モードほどではない．一方で，判定結果の通知の時間差の問題はない．

したがって，解答のバリエーションが多い場合や，評価事例が豊富でない場合にはこのモードにするのが良いと思われる．

(3) **動作のみ評価モード**

プログラムの動作評価の結果，動作が正しければ，合格とするモードである．実現方法を問わないような題意の問題に対して用いるとよい．

5.3 システム構成

実現したシステムはサーバ，学生用クライアント，教員用クライアントから構成される．

(1) **学生用クライアント**

学生用クライアントは，CASL プログラムの編集，アセンブル，2 モードのシミュレート機能を持つシミュレータ WCASL☆に，プログラム提出機能と，プログラムの動作評価機能を追加したものである．これは，Windows アプリケーションとして実現した．

(2) **教員用クライアント**

教員用クライアントは，WWW ブラウザでパスワードによってセキュリティのかかったページにアクセスすることで実行する．したがって，教員は，ウェブベースのインタフェースで，提出状況の閲覧機能，判定入力支援機能，提出されたプログラムの閲覧機能を実行できる．提出状況の閲覧ページには，教員の評価が必要なプログラムがマークアップされ，それをクリックすることで，判定入力支援ページが呼び出せる．判定入力支援ページでは，学生のプログラムと事例のプログラムを見ることができ，アドバイス欄にアドバイスを書き込み，「合格」か「不合格」かをマウスで選択して実行ボタンをクリックすると，学生に自動的に電子メールが送られる．これは，ディスクに保存された

☆ http://www.ics.teikyo-u.ac.jp/wcasl/から取得可.

ファイルを開いては評価を行うという操作に比較して格段に使いやすい．これらの機能は，サーバに CGI プログラムとして実現した．

(3) **サーバ**

サーバは，問題情報，事例，一般化ルール等のプログラム評価に必要な知識，提出されたプログラム，学生の提出状況データを保持し，クライアントにおける操作のログを記録する．また，事例に基づくプログラムの実現方法の評価や教員用クライアントのための機能は，以下のような CGI プログラム群として実現した．

(a) 提出受付用 CGI：提出を受け付け，提出状況データを更新する．自動モードの場合は，事例に基づく実現方法の評価機能を実行し，判定可能なら，自動的に電子メールを送信する．

(b) 判定入力支援用 CGI：事例に基づく実現方法の評価機能を実行し，システムの判定結果を提示するとともに，判定結果とアドバイスを入力するためのフォーマットを提供する．

(c) 判定入力確認用 CGI：(b) で入力された判定結果とアドバイスについて，それでよいかの確認をとる．

(d) 結果送付および事例登録 CGI：判定結果とアドバイスを電子メールで送信する．また，事例登録の必要がある場合に，提出されたプログラム，教員の入力した判定結果，およびアドバイスを新事例として追加する．

(e) 提出状況閲覧用 CGI：上で述べた提出状況一覧を提示する．

(f) 提出プログラム閲覧用 CGI：個々のプログラムを提示する．

6. 実現したシステムの評価

6.1 実験条件

実現したシステムの判定精度と教員の作業軽減の効果を評価することを目的として，本学情報科学科 2 年生を対象とした演習授業の 2 つのクラス（仮に A 組，B 組と呼ぶ）において，本システムを実際に使用した．授業を履修した学生の人数は，A 組が 79 人，B 組が 73 人である．全員が合格となるまで，プログラムの再提出を行うので，プログラムの提出数は，学生数よりも多くなる．実際にシステムを使用した際に出題した問題を**表 1** に示す．

システムの判定精度は，システムが判定を行ったプログラムのうち，システムと教員の判定結果が等しいケースの割合で評価する．また，教員の作業軽減の効果を精確に評価することは難しいが，おおよそのところは，以下のような値で評価が可能である．

表 1 出題した問題
Table 1 Problems given to students.

問題名	問題の説明	対象
P1	2 つの値の大きい方を求める	B 組
P2	N 個のデータの加算	B 組
P3	N 個のデータの最大値を求める	B 組
P4	N ビットの循環右シフト	B 組
P5	N ビットの循環左シフト	A 組
P6	スタックを使って () の対応検査	B 組
P7	スタックを使って文字列反転	A 組

表 2 問題 P3 に対して提出されたプログラムの判定結果の比較
Table 2 Comparisons of the system and a teacher in their judgements on the acceptability of programs submitted as answers of P3.

	教員の判定		
システムの判定	合格	不合格	合計
合格	60 (59)	0 (0)	60 (59)
不合格	2 (0)	15 (13)	17 (13)
判定不可	14	5	19
小計	76	20	96
動作評価で不合格	0	129 [44]	129 [44]
合計	76	136 [64]	212 [140]

( )：学生のプログラムに事例が完全照合したケースの値．
[ ]：1 人の学生が動作評価で不合格になるプログラムを多くて 2 個提出したと仮定した場合の値

- 提出プログラムのうち，システムが自動的に判定を行えたプログラム数の割合（軽減量 1）．
- システムが自動的に行えた評価タスク数の割合（軽減量 2）：プログラムの評価作業が，(a) 動作の評価，(b) 実現方法の評価，(c) アドバイスの作成の 3 つのタスクからなると仮定し，全体のタスク数に対するシステムが行うことができた評価タスク数の割合．タスク数は，(a) と (c) は提出されたプログラム数，(b) は動作が正しかったプログラム数で，全体ではそれらの合計とした．

6.2 実験結果と考察

表 1 のうちの P3 に関して，提出されたプログラムに対するシステムと教員の判定結果の比較を**表 2** に示す．丸括弧内の数は，システムの判定で事例が完全照合をした場合の数を表す．本システムによる提出では，教員に直接提出する場合に比較して，学生は “気軽に” 提出するため，プログラムの動作を自分で十分に確認せずに提出する学生が多くなると予想される．これを補正するために，1 人の学生が提出した，動作評価で不合格になるプログラムを多くて 2 個に限定すると，提出数の合計はそれぞれ [ ] 内の値になる．以下の計算ではこれらの値を用いる．

(1) 判定精度：表 2 から，システムが判定を行ったプログラム数は 140 から判定不可の 19 を引いた 121

表 3 システムの評価結果
Table 3 The results of the system evaluation.

	P1	P2	P3	P4	P5	P6	P7
事例数	15	10	29	34	38	34	40
命令数	9	10	15	19	20	26	20
精度	100	100	98	100	100	100	100
軽減量 1	89	90	83	78	77	74	73
軽減量 2	93	89	86	78	80	79	74

であり，そのうちの 2 個を除いて教員の判定と一致している．このことから，判定精度は約 98%となる．また，判定が一致しなかった 2 つのケースは完全照合ではなかったので，事例が完全照合した場合に限ると，判定精度は 100%である．

(2) システムが自動的に判定を行えたプログラム数の割合（軽減量 1）：表 2 のうちで，事例が完全照合した 59+13 ケースと，動作評価で不合格となった 44 ケースについては，教員は評価を行う必要がない．これらの提出総数（140）に対する割合は約 83%である．

(3) システムが自動的に行えた評価タスク数の割合（軽減量 2）：表 2 における全体のタスク数は，(a) 動作の評価が 140，(b) 実現方法の評価が 96，(c) アドバイスの作成が 140 で合計 376 である．これらのうち，教員が行った評価タスクは (b) が 23，(c) が 28（判定不可のケースとシステムが生成したアドバイスがそのまま使えなかったケースの合計）で合計 51 であった．したがって，325 のタスクをシステムが行ったと考えられるので，全体のタスク数に対する割合は約 86%である．

表 1 の問題に対して同様に計算した結果を**表 3** にまとめた．ここで，事例数は事例ベースに最終的に登録された事例数であり，命令数は事例プログラムの命令数の平均である．これらの値は，問題に対する解答のバリエーションの多さを表すものと考えられる．

6.2.1 システムの判定結果の精度に関する考察

システムの判定精度に関しては，P3 で 98%のほかは 100%であり，十分に高い精度である．システムと教員の判定が異なったのは，表 2 で示した 2 ケースのみであり，これらは減算命令の有無が合否のポイントとなった．

完全照合のケースに限ると，判定精度はすべての問題で 100%であった．4.2.1 項で述べたとおり，完全照合の場合は評価対象は事例とほとんど同一と見なせるので，完全照合のケースで判定精度が 100%となるのは予想された結果である．

6.2.2 作業負荷の軽減効果に関する考察

表 3 の結果から，教員の評価作業は，簡単な問題（P1 や P2）で 90%程度軽減され，やや複雑な問題（P4〜P7）でも 80〜70%程度軽減されている．解答のバリエーションが多くなるほど，軽減量も少なくなる傾向があるが，初等アセンブラプログラミング教育で用いる程度の問題において，本システムによる教員の評価作業軽減の効果は顕著である．

また，作業負荷軽減に関しては，ほかに以下のような効果もある．

- 学生が提出するすべてのプログラムについて動作の評価が自動化されるため，この作業は 100%軽減され，見落としもなくなる．
- ウェブページで簡単に学生のプログラムにアクセスでき，教員の評価が必要なプログラムがマークアップされるので，評価対象にアクセスする作業負荷が軽減される．
- 入力した判定やアドバイスが自動的に電子メールで送付されるので，学生に結果やアドバイスを通知する作業が軽減される．

以上のような考察から，教員の評価作業負荷軽減に関して，本システムの効果はきわめて大きいといえる．

6.3 問題点と今後の課題

(1) 事例の一般化：本システムの判定精度は完全照合のときには 100%を達成した．ただし，事例のプログラムリストは一般化表現をとるので，事例が過度に一般化されると，完全照合でも教員の判定とシステムの判定が一致しないこともありうる．一般的に 100%の判定精度を達成するためには，事例一般化の際に，過度な一般化☆を防ぐ必要がある．

(2) 事例の選択基準：本システムでは，4.2.1 項で述べたとおり，「順序関係の異なること」を「冗長命令の有無」よりも優先させて適用する事例を選択した．しかし，今回の使用でシステムと教員の評価の違ったケースでのポイントは，冗長命令の有無であった．そこで，完全照合でない場合の事例の選択基準を再吟味する必要がある．

(3) 作業負荷軽減：本システムの導入により，問題情報や初期事例の作成作業が増える．これらの作業の増加は評価作業の軽減効果に比較すると非常に小さいが，必要な情報を一定のフォーマットに従って記述するには，ある程度の慣れが必要である．そこで，問題情報や初期事例の作成ツールが課題となる．

☆ たとえば，我々は，算術比較命令（CPA）と論理比較命令（CPL）の両方をカバーする一般化命令 COMP を定義した．教員が「論理演算なので論理比較を行わせたい」という意図を持っているときに，この一般化命令を使って一般化を行うと，合格と不合格の両方のプログラムと照合してしまう．このような状況を過度な一般化と呼ぶ．

(4) アドバイス文の修正機能：本システムでは，4.2.3 項で述べた簡単なアドバイス文の修正機能を実現したが，現状の機能は貧弱であり，それがアドバイス文をそのまま使用できない原因の 1 つになっている．強力なアドバイス文修正機能の実現が課題となる．

(5) 冗長命令の検出：本システムでは，1 命令ずつ削除してプログラムの動作を確認することで，冗長命令を検出するので，(a) 2 命令のどちらか一方だけが削除可能なときでも，その両方が冗長命令として指摘される，(b) テストデータによっては，本来冗長でない命令を冗長命令と指摘することがある，等の問題がある．ただし，そもそもすべての冗長命令が有害とは限らないので，教員はシステムの検出結果を吟味したうえでアドバイスをするべきである．

(6) 事例管理：教員は，同じプログラムに対しても，アドバイス文の表現をより良く変えることがある．これがアドバイス文をそのまま使用できない第 2 の原因である．つまり，一度登録した事例の内容に変更が生じうるため，適切な事例管理を行う必要がある．その方法論を確立することも今後の課題である．

7. むすび

提示した問題に対して学生が作成したプログラムの評価を支援するシステムの実現方法を提案し，その方法に基づいてプログラム評価支援システムを実現した．また，実現したシステムを実際の授業で用いた結果，本システムによる教員の評価作業負荷軽減の効果はきわめて大きいことが明らかになった．

本研究で提案した実現方法に基づくシステムは，教室内の授業だけでなく，ネットワークを利用した遠隔授業におけるプログラム評価支援ツールとしても利用可能であり，幅広く役立つものと考えられる．

謝辞 実現システムについてご意見をいただき，また，実際の授業での実験にご協力いただいた本学技術職員の高井久美子さん，システム開発にあたり，熱心に協力された本学卒研生矢古宇努君，半田博美君に感謝する．なお，本研究の一部は文部省科学研究費補助金（11680400）の援助による．

参考文献

1) 渡辺博芳，荒井正之，武井惠雄：CPU とアセンブラ授業のための合否判定支援システム，情報処理学会研究報告，コンピュータと教育，Vol.98, No.48, pp.61–68 (1998).
2) Watanabe, H., Arai, M. and Takei, S.: Automated Evaluation of Novice Programs Written in Assembly Language, *Proc. ICCE99*, Vol.2, pp.165–168 (1999).
3) 渡辺博芳，荒井正之，武井惠雄：CPU とアセンブラ授業のための事例に基づくプログラム評価支援システム，情報処理学会研究報告，コンピュータと教育，Vol.99, No.54, pp.33–40 (1999).
4) 藤原祥隆，松西年春，岡田信一郎，大鎌 広，後藤寛幸，黒丸鉄男：プログラミング演習支援のための階層分散処理システムの設計と評価，電子情報通信学会論文誌，Vol.J78-D-II, No.11, pp.1701–1709 (1995).
5) 吉野 孝，宗森 純，伊藤士郎，長澤庸二：教育用プラットフォーム DEMPO II の開発とプログラミング演習への適用，情報処理学会論文誌，Vol.37, No.5, pp.891–901 (1996).
6) 高橋参吉，松永公廣：プログラミング学習のための電子学習環境の構築，日本教育工学会論文誌，Vol.23, No.3, pp.155–165 (1999).
7) 福島 学，浮貝雅裕，菅原研次，城戸健一：プログラミング演習のためのハイパテキスト型教材の実装，情報処理学会論文誌，Vol.34, No.6, pp.1246–1257 (1993).
8) 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).
9) Adam, A. and Laurent, J.P.: LAURA, A System to Debug Student Programs, *Artificial Intelligence*, Vol.15, pp.75–122 (1980).
10) Murray, W.R.: Automatic Program Debugging for Intelligent Turoring Systems, *Computional Intelligence*, Vol.3, pp.1–16 (1987).
11) Johnson, W.L.: Understanding and Debugging Novice Programs, *Artificial Intelligence*, Vol.41, pp.51–97 (1990).
12) Ueno, H.: Concepts and Methodologies for Knowledge-Based Program Understanding – The ALPUS's Approach, *IEICE TRANS. INF & SYST*, Vol.E78-D, No.2, pp.1108–1117 (1995).
13) Ueno, H. and Inoue, T.: A Shared Intelligent Programming Environment on the Internet for Learning C Programming, *Proc. ICCE99*, Vol.1, pp.752–759 (1999).
14) 海尻賢二：ゴール/プランに基づく初心者プログラムの認識システム，電子情報通信学会論文誌，Vol.J78-D-II, No.V2, pp.321–332 (1995).
15) Kim, S. and Kim, J.H.: Algorithm Recognition for Programming Turoring Based on Flow Graph Parsing, *Applied Intelligence*, Vol.6, Iss.2, pp.153–164 (1996).
16) Kaijiri, K. and Sekimoto, R.: Program Diagnosis System on World Wide Web, *Proc. ICCE99*, Vol.1, pp.729–735 (1999).
17) Konishi, T., Suyama, A. and Itoh, Y.: Evalu-

ation of Novice Programs Based on Teacher's Intention, *Proc. ICCE95*, pp.557–566 (1995).
18) 服部徳秀，石井直宏：プログラミング演習の評価サポートシステムの構築，教育システム情報学会誌，Vol.14, No.1, pp.21–28 (1997).
19) 関本理佳，海尻賢二，山形昌也：ネットワークを利用したレポート受付・評価支援システムの実現，教育システム情報学会誌，Vol.14, No.5, pp.217–222 (1998).
20) Kolodner, J.: Case-Based Reasoning, Morgan Kaufmann Publishers (1993).
21) Leake, D. (Ed.): Case-Based Reasoning: Experiences, *Lessons and Future Directions*, AAAI Press/MIT Press (1996).

付録 本システムで用いた一般化命令

現時点で定義されている一般化命令を以下に示す．
(1) SET ?RG1 ?C1：レジスタの設定．LD 命令と LEA 命令をカバーする．
(2) INC ?RG1 ?C1：レジスタの値を増やす．ADD 命令と LEA 命令で指標レジスタを使ってレジスタの値を増やす記述をカバーする．
(3) DEC ?RG1 ?C1：レジスタの値を減らす．SUB 命令と LEA 命令で指標レジスタを使ってレジスタの値を減らす記述をカバーする．
(4) COMP ?RG1 ?L1：比較．CPA 命令と CPL 命令をカバーする．
(5) J+ ?L1：フラグレジスタ（FR）が正のとき分岐．JMI 命令，JZE 命令と JMP 命令を組み合わせた記述をカバーする．
(6) j- ?L1：FR が負のとき分岐．JMI 命令単独の記述と JPZ 命令と JMP 命令を組み合わせた記述をカバーする．
(7) J0 ?L1：FR が零のとき分岐．JZE 命令単独の記述と JNZ 命令と JMP 命令を組み合わせた記述をカバーする．
(8) J+- ?L1：FR が零でないとき分岐．JNZ 命令単独の記述と JZE 命令と JMP 命令を組み合わせた記述をカバーする．
(9) J+0 ?L1：FR が正か零のとき分岐．JPZ 命令単独の記述と JMI 命令と JMP 命令を組み合わせた記述をカバーする．
(10) J-0 ?L1：FR が負か零のとき分岐．JMI 命令と JZE 命令を組み合わせた記述をカバーする．
(11) J+0OR+ ?L1：分岐命令．一般化命令である J+0 命令と J+命令をカバーする．
(12) J+0OR0 ?L1：分岐命令．一般化命令である J+0 命令と J0 命令をカバーする．
(13) J+-OR+ ?L1：分岐命令．一般化命令である J+-命令と J+命令をカバーする．
(14) J+-OR- ?L1：分岐命令．一般化命令である J+-命令と J-命令をカバーする．
(15) J-0OR- ?L1：分岐命令．一般化命令である J-0 命令と J-命令をカバーする．

(平成 12 年 2 月 14 日受付)
(平成 12 年 11 月 2 日採録)

渡辺 博芳（正会員）
1963 年生．1986 年宇都宮大学工学部情報工学科卒業．1988 年同大学院工学研究科修士課程修了．工学博士．栃木県工業技術センターを経て，1991 年より帝京大学理工学部情報科学科助手．知識工学，事例ベース推論，機械学習，ニューロコンピューティング等に興味を持つ．2000 年情報処理学会第 60 回全国大会大会優秀賞．電子情報通信学会，人工知能学会，AAAI 各会員．

荒井 正之（正会員）
1958 年生．1981 年東京理科大学理工学部経営工学科卒業．1995 年宇都宮大学大学院工学研究科生産・情報工学専攻博士後期課程修了．工学博士．1988 年帝京大学理工学部情報科学科助手，現在同大学専任講師．ニューロコンピューティング，文字認識，自然言語理解，知識工学等に興味を持つ．著書「文字認識技術」（トリケップス社）等．電子情報通信学会，IEEE Computer Society 各会員．

武井 恵雄（正会員）
1938 年生．1961 年東北大学理学部天文及地球物理学科第二卒業．1963 年同大学院理学研究科地球物理学専攻修士課程修了．理学博士．東北大学理学部助手，同大学情報処理教育センター助教授を経て，1992 年帝京大学理工学部情報科学科教授．力学系，知的信号処理，分散環境における協調学習環境の実現，情報教育の哲学的側面等に興味を持つ．地球電磁気・地球惑星圏学会，日本教育工学会各会員．