コンピュータと教育 48－6
（1998．5．29）

CPUとアセンブラ授業のための合否判定支援システム

渡辺 博芳　荒井 正之　武井 惠雄

帝京大学 理工学部 情報科学科

本論文では，COMET/CASLを教材としたCPUとアセンブラ授業において，提示した課題に対して学生が作成したプログラムの合否判定を支援するシステムの実現方法を提案する．ネットワーク環境に構築された学習者のプログラムを判定するシステムは，通常の演習授業においてはもちろん，宿題の提出や遠隔授業における合否判定支援ツールとして用いることも可能である．プログラムの合否判定処理は，プログラムの動作の評価とプログラムの認識の2つの処理によって行う．動作評価処理ではあらかじめ用意した複数のテストデータに対する動作を評価する．一方，プログラムの認識処理では一般化表現で記述された解答例プログラムと学習者のプログラムの照合を行う．本論文ではこのような方針に基づく合否判定支援システムの実現方法の詳細を述べる．本システムを用いて少人数の授業を実験的に行ったところ，授業での実用が可能であり，教員の合否判定タスクの負荷を大幅に減少できる見通しを得た．

An Evaluating System for the Students' Programs of the CPU and Assembler Course

Hiroyoshi Watanabe, Masayuki Arai and Shigeo Takei

Dept. Information Sciences
School of Science and Engineering, Teikyo University

In this paper, we propose a method of implementing the system which supports the evaluation of students' programs for the CPU and Assembler course. We use COMET and CASL as teaching materials in the course. The system would be useful for not only classes but also submitting assignments, because it is implemented in the computer network environment. The process of the program evaluation consists of two sub-processes: first, actions of the program are tested with several pairs of data, and then the program is recognized by matching it against correct programs which are represented in generalized forms of CASL instructions. The system was utilized for an experimental class and the results of the experiments show that the system must be practical and reduce teachers' loads of the evaluation drastically.

1 まえがき

本学の情報科学科において，CPUの動作の徹底理解とアセンブラ・プログラミングの基礎の習得を目的として，COMET/CASLを教材に"CPU/Assembler"演習授業を行っている．典型的な授業は，まず，ある概念や命令，アルゴリズムなどについて講義や説明をした後，それらを応用した課題を提示し，その課題のプログラムを授業時間内(あるいは授業日内)に完成させるといった形式である．課題の合否は，教員が学生1人1人と面談をしながら判定する．この授業の間，教員は(a)学生からの質問に対する回答，(b)学生のプログラムの合否の判定，(c)理解度や進捗状況の把握を目的とした学生への質問などを行う．また，ある事柄について多くの学生が理解不足と判断した場合には，学生の作業を一時中断させて，全体への説明を行う．このように，授業における教員のタスクは多い．また，課題の合否判定は1人ずつ行われるので，ピーク時には判定待ちの行列が生じるという問題もある．そのため，ややもすると教員は課題の合否判定に注力するあまり，理解が不足している学生に対するフォローがおろそかになるケースも少なくない．そこで，より質の高い授業をより効率的に行うために，授業を支援するためのツールが望まれる．

支援を行うタスクとして，講義や説明の支援，学生のプログラミング作業の支援，理解不足の学生への支援などが考えられるが，我々は合否判定のタスクを支援することが最も効果的であると考えた．なぜならば，合否判定を支援することにより，教員の合否判定のタスクの負荷が軽減することでより効率的な授業が行える．また，教員は理解不足の学生のフォローや学生の理解度の把握などの他のタスクにより多くの時間を使えるので，授業の質も向上することが期待される．さらに，学生にとっても合否判定の待ち行列から開放されることで，より有効に時間を使えるという効果もある．そこで本研究は，提示した課題に対して学生が作成したプログラムを分析し，合否の判定を支援するシステムの実現方法を提案することを目的とする．

学生のプログラムの合否判定を行うためには，学生のプログラムを認識する必要がある．プログラム理解やプログラム認識の研究は古くから行われており，これらの研究は，主にreverse engineeringとプログラム教育支援の2つの観点に分けられる．プログラム教育支援では，LAURA[1]，PROUST[2]，ALPUS[3]，PascalRecognizer[4]などのシステムがある．これらのシステムでは，学生のプログラムをデバックし，学生に誤りを示唆することが主な目的となっている．それに対して，本研究で提案する手法では学生のプログラムが正しいプログラムであるかどうかの認識を主な目的とし，正しくない場合のバグの所在などは解析しない．また，上にあげたシステムのプログラム理解のアプローチには実現方法の詳細な違いはあるものの，抽象化されたプログラムのパタン記述と入力されたソースコードとの照合を行っている．本手法のプログラムの認識においてもそれらと同様なアプローチをとる．さらに，プログラムの合否判定を行う上では，そのプログラムが正しく動作することが重要であるので，本手法では合否判定においてプログラムの動作の評価も行う．

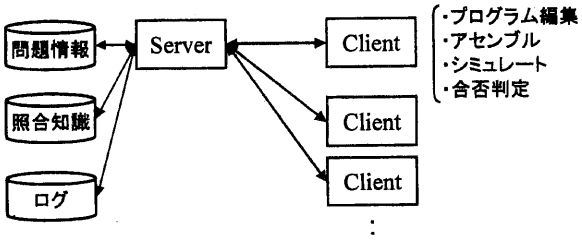

図1: システム構成

2 システムの概要

2.1 システム構成

構築しようとするシステムの構成を図1に示す．学生が使用するクライアントパソコン上には，CASLプログラムの編集，アセンブル，シミュレーション，合否判定の機能を持つシステムを実現する．一方，サーバには合否判定に用いるための問題情報，照合知識を蓄え，クライアントの要求に応じてこれらの情報を提供する．また，学生のクライアントシステムの操作や合否判定の過程に関するログを収集して記録する．

本学の演習では，著者の1人が開発したCASLとCOMETのシミュレータWCASL[5]を用いてい

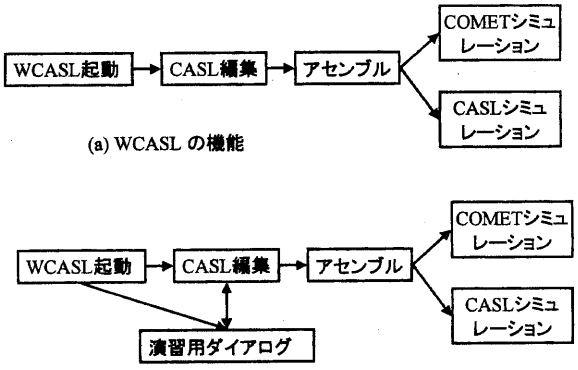

図 2: クライアントシステムの機能

る．WCASL は図2(a) に示すように，CASL プログラムの編集，アセンブル，2モードのシミュレーション機能を持つ．本研究におけるクライアントシステムは，図2(b) に示すように，WCASL に演習用ダイアログウィンドウを追加することで実現する．演習用ダイアログには提示されている問題文の閲覧，プログラム表示の機能があり，それにより，学生が問題とそれに対するプログラムを確認した上で，合否判定要求ボタンによって合否判定を行う．

2.2 合否判定の基本方針

プログラムの合否を判定する上で，まず，重要になるのがプログラムが正しく動作するか否かであろう．また，演習では，ある概念，アルゴリズム，命令の使い方などの習得を目的として課題を与える．従って，例えば「スタックを用いて」などのようにプログラムの実現方法に制限を与えることが多い．そこで，プログラムの実現方法についても評価する必要がある．これらのことを考慮して，本システムでは合否判定の処理をプログラムの動作の評価とプログラムの認識の2つのフェーズに分けて行うこととした．合否判定処理の流れを図3に示す．まず，プログラムの動作評価を行い，誤りがあると思われる場合は不合格とする．動作評価をパスしたプログラムについて認識を行い，プログラムの認識に成功すれば合格とする．プログラムの動作が正しいが，認識に失敗した場合は仮合格とし，教員が合否判定を行う．

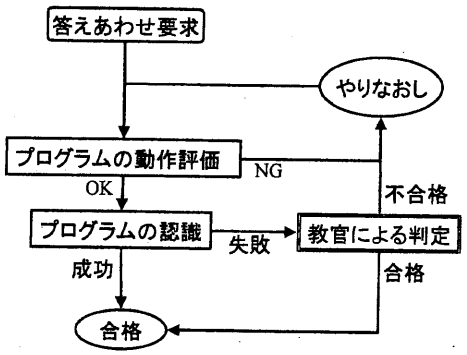

図 3: 合否判定処理の流れ

プログラムの動作評価では，あらかじめ用意しておいた複数のテストデータについて，学生の作成したプログラムを動作させて，全てのデータに対して正しく動作すれば合格とする．

プログラムの認識では，あらかじめ用意しておいた解答例プログラムの一般化表現との照合を行い，照合が成功すれば合格とする．解答例プログラムは複数用意することもあるが，その場合，照合がとれる解答例プログラムが1つでも存在すれば合格となる．解答例プログラムを一般化しておくことにより，実現されたプログラムにある程度の多様性があっても，比較的少数の解答例を用意すればよいことになる．一般化表現である解答例との照合において，一般化表現からCASL の命令に対応付けを行うプロダクションルールを用いる．このルールの集合を照合知識と呼ぶ．また，以後，解答例として用意した一般化表現によるプログラムを解答例プログラム，あるいは単に解答例，学生が作成した認識の対象となるプログラムを認識対象プログラム，あるいは認識対象と呼ぶ．解答例プログラムと認識対象プログラムの照合のアプローチはいくつかあると考えられるが，本研究では，まず，解答例の各命令についてそれに照合し得る認識対象の命令の候補を生成し，それを絞り込むことによって照合を行うアプローチをとる．

2.3 問題情報

問題情報は，課題として提示する問題に関する以下の情報を定められた形式に従って，1つの問題について1つのテキストファイルに記述する．

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．すなわち，データは例えば以下のように保存されている．

```
N       DC    5    ;データの個数
DATA    DC    1
        DC    2
        DC    3
        DC    4
        DC    5
```

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

図 4: 問題情報の例

- 問題識別子
- 問題名
- キーワード
- 問題文
- ラベル情報
- 動作評価のためのテストデータ
- プログラム認識のための解答例

問題識別子は全ての問題情報においてユニークな識別子である．問題名は，問題1とか，問題4-2などのような授業においてその問題を参照する名称である．すなわち，将来的に問題情報をデータベース化することを想定しており，そのときの識別子が問題識別子であり，問題名は授業での名称に合せて変更可能とする．また，データベース化した際に役立つように，その問題の特徴を表すキーワードを複数与えておく．問題文は，問題を表現したテキストである．ラベル情報はテストデータと解答例で用いられているラベルに関する情報で，ラベル名とラベルの説明から成る．動作のためのテストデータとプログラム認識のための解答例に関しては後で述べる．テストデータと解答例部分を除いた問題情報の例を図4に示す．

入力：

```
N       1       3
DATA    3       -5  3  0
```

出力：

```
SUM     1       -2
```

図 5: テストデータの例

3 プログラムの動作評価

3.1 テストデータの形式

プログラムの動作評価を行うために，入力データと出力データの組合わせで1組のテストデータを構成する．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の列である．1組のテストデータの例を図5に示す．図5はラベルNのサイズが1でそこには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．図5では入力データが2つ，出力データが1つであるが，出力データが複数のケースも扱える．このようなテストデータの組を複数用意する．

3.2 動作評価処理

動作評価処理は，ラベルの対応情報の取得を行った後，テストデータの数だけ，データ部の書換え，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．

(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4で示すラベル情報を用いて，例えば次のような質問をする．

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？

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わかる．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，質問には，ある課題に対する最初の合否判定において答えればよい．同じ課題について再度合否判定を行う場合は，ラベル情報の修正を行うか否かの確認があり，修正しなければ以前の情報を用いる．

(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．

(i) 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．

(ii) 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．

(iii) 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．

全てのテストデータについて正しく動作すれば，合格とする．1つでも正しく動作しないテストデータがあれば，不合格である．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 プログラムの認識

4.1 解答例と照合知識

(1) 解答例

解答例プログラムはCASLプログラムを一般化した表現をとる．その形式は，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．汎用レジスタのうち，指標レジスタとして使えるレジスタをGRXn，必ずしも指標レジスタでなくともよい場合はGRGnという形式で表現する．nは1以上の整数であり，同じ役割をもつレジスタに対して同じ値を割り当てる．この数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を超える値が用いられることもある．また，一般化表現として新しく導入する命令の形式は照合知識において定義する．照合知識については後で述べる．図4の問題に対する解答例プログラムの例を図6に示す．ここで，SET，INC，COMP，J+−OR−は一般化表現として導入した命令である．

```
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      5
DATA    DC      1
        DC      2
        DC      3
        DC      4
        DC      5
        END
```

図 6: 解答例プログラムの例

```
SET   ?RG1  ?C1  →       LEA   ?RG1, ?C1

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

図 7: 照合知識の例(SET)

(2) 照合知識

照合知識は，一般化表現をCASLの命令に変換するためのプロダクションルールの集合である．図7に新しく導入したSETという命令に関する照合知識(2つのプロダクションルール)を示す．?で始まる記号は変数を表している．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命令により定義された値を設定する方法がある．図7の上が前者，下が後者である．

プロダクションルールの右辺部，すなわち，通常一般化表現の命令を実現するためのCASL命令列が記述される部分に，一般化表現の命令を記述することも可とする．その際，右辺部に記述され

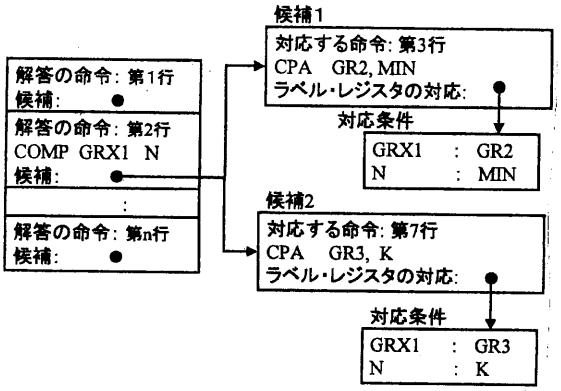

図 8: ワーキングリストの例

解答例の記号	認識対象の候補	対応記号
GRX1	GR2, GR3	
N	MIN, K	

図 9: ラベルとレジスタの対応表の例

る一般化表現の命令は既に定義されていること，ルールの適用においてループを生じさせないことが前提となる．

4.2 認識処理のためのデータ構造

プログラム認識を行うために図8に示すような命令の照合のためのワーキングリストと図9に示すようなラベルおよびレジスタの対応表を用いる．

ワーキングリストは解答例プログラムの命令(行)と対応する認識対象プログラムの命令(行)の候補とその候補が対応するためのラベルおよびレジスタの対応条件を表す．図8の例では解答例プログラムの第2行に対して2つの候補が示されている．すなわち，1つめの候補は認識対象の第3行であり，そのときGRX1にGR2，NにMINがそれぞれ対応することが条件である．また2つめの候補は認識対象の第7行であり，その条件はGRX1にGR3，NにKがそれぞれ対応することである．図8の例では解答例の1つの命令に認識対象の1つの命令が対応しているが，解答例の1つの命令に認識対象の複数行が対応することもある．

一方，対応表はプログラム認識全体において，解答例プログラムと認識対象プログラムの間で対応するレジスタおよびラベルについての候補と最終的に決定された対応記号を管理する．

4.3 プログラム認識処理

プログラム認識処理は，(1) ラベルとレジスタの対応表の初期化，(2) ワーキングリストの初期化(候補の生成)，(3) 候補の絞り込みと決定の順に行う．

(1) 対応表の初期化

解答例プログラムのラベルとレジスタを対応表に登録し，プログラムの動作評価で取得したラベルの対応情報を対応記号の欄に反映させる．

(2) ワーキングリストの初期化

解答例プログラムの各命令(各行)に対して，候補となり得る認識対象プログラムの命令を求め，それらの照合の条件(ラベルやレジスタの対応)を生成する．ラベルやレジスタの対応条件のうち，図9に示す対応表の候補の欄に含まれていないものを追加する．候補となる命令は以下のようにして求める．

(i) 解答例プログラムの命令がCASLの命令であるときは 命令コード が等しく，かつ指標レジスタの有無が等しい認識対象の命令を候補とする．

(ii) 解答例プログラムの命令が一般化表現であるときは，まず，照合知識を適用して，命令列を得る．その命令列に一般化表現が含まれる場合は，さらに照合知識を適用する．これにより，複数組の命令列が生成される．次にそれらの命令列の組について以下の処理を行う．

- 命令列の各命令に対して，CASL命令の場合と同様に，命令コード が等しく，かつ指標レジスタの有無が等しい認識対象の命令を割り当てる．
- 命令列の全ての命令に対応する認識対象の命令が割り当てられたら，候補とする．
- 命令列の先頭を進めて，上の2つの処理を行い，全ての候補を生成する．

(3) 候補の絞り込みと決定

候補の絞り込みと決定の処理を図10に示す．ワーキングリストの全てのエントリについて候補が1つになれば照合は成功となる．ワーキングリストにおいて候補が1つに決まると，その対応付けのための条件は成立するはずなので，対応表を更新できる．さらに，対応表においてラベルやレジ

- 照合の成否が決まるまで以下の処理を行う
  - 命令の候補数が0なら，照合失敗
  - 命令の候補数が1なら，それに決定し，対応表を更新
  - 命令の候補数が2以上なら，以下の処理を行う
    - 対応表の情報を基にワーキングリストの候補を絞り込む
    - ワーキングリストの候補の情報を基に対応表の候補を絞り込む
  - 対応表の候補数が1なら，それに決定する
  - 対応表とワーキングリストのいずれも更新されなければ出現順序に基づいてワーキングリストの命令の対応を1つ決定する

図 10: 候補の絞り込みと決定処理

スタの対応が決まると，ワーキングリストの候補において対応づけのための条件が成立しないものを候補から削除できる．このように，ワーキングリストと対応表の情報を用いてお互いに候補を絞り込む．従って，ワーキングリストと対応表のどちらも更新されない場合は，それ以上絞り込みが行われない．そこで，命令の出現順序に基づいてワーキングリストのエントリの1つを決定し，再びワーキングリストと対応表の絞り込みの処理を行う．

出現順序に基づく命令の対応の決定は以下の処理の記述順に試み，ワーキングリストの命令の対応が新たに1つ決定した時点で終了する．

(i) 対応を決めようとする命令の直前の命令と直後の命令の対応が決定しており，その対応先に挟まれた候補命令が存在すれば，それに対応させる．

(ii) 対応を決めようとする命令の直前の命令の対応が決定しており，直前の命令が対応している命令の直後に候補があれば，それに対応させる．または，対応を決めようとする命令の直後の命令の対応が決定しており，直後の命令が対応している命令の直前に候補があれば，それに対応させる．

(iii) 対応が決定していない命令の最初のものを，その候補のうちの最初のものに対応させる．

5 実験

5.1 実験条件

これまで述べた手法に基づくクライアントシステムをVisual C++で作成した．本学の計算機演習室(CL室)は，サーバマシン7台に Windows NT Serverを搭載し，クライアントにはWindows NT Workstation を搭載している[6]．図1におけるサーバに対応するシステムも作成するのが望ましいが，今回の実験ではWindows NT Server のフォルダからネットワークを介して問題情報や照合知識を読み込み，同様にログファイルを書き込むことにした．

研究室の4年生14人を対象にして，2年生で行われているCPUとアセンブラの授業の縮小版のような形でゼミを実施した．この14人の中には2年生でこの授業を履修した学生と履修していない学生が含まれる．まず，簡単な問題を用いて合否判定支援システムの使い方のチュートリアルを行い，以下の2問を提示した．

- 問題1： 2つのラベルの内容のうち，大きい方を別のラベルに格納するプログラム．
- 問題2： 連続するメモリ領域に存在するデータの合計を求めるプログラム．

問題1については6組のテストデータと4つの正解プログラムを，問題2については2組のテストデータと3つの正解プログラムをそれぞれ用意した．ゼミは1時間40分程度で打ち切り，最後に学生に感想を書いてもらった．

表 1: プログラム評価の解析結果

	教員の判定		
システムの判定	合格	不合格	合計
合格	12	1	13
仮合格	6	0	6
不合格	0	7	7
アセンブルエラー	0	1	1
ハングアップ	0	1	1
合計	18	10	28

5.2 実験結果

時間内に2問とも完成させた学生は5人，どちらか1問を完成させた学生が6人，1問も完成できなかったのが3人であった．ログを解析した結果，全く同じプログラムに対する合否判定要求を1つに数えると，2問について延べ28回の合否判定要求があった．それらの解析結果を表1に示す．1人

の学生が1つの問題に対して，2種類以上のプログラムを作成して2つとも合格しているケースもあるので，表1の合格者と上で述べたプログラムを完成させた学生の数は一致しない．

表1でシステムの判定が合格で教員の判定が不合格であるケースは，プログラムの最後にEND命令が書かれていなかった．これはWCASLでアセンブル処理をする際，END命令は単に削除するだけであるためで，WCASLの処理系を改善することで対処できる．また，合否判定を行ったところ，システムがハングアップしたケースは，EXIT命令が抜けていたため，データ領域を命令として実行したために生じたもので，無限ループ時の処理や定義されていない命令の実行時の処理を改善することで対処できる．これらの他は適切に判定が行われたと言える．ハングアップのケースを除くと，合否判定要求のあった27ケースのうち，6ケースを除いてシステムが自動的に判定を行ったので，教員の合否判定タスクにおける負荷は22%程度になり，効率化が実現できることが明らかになった．

また，システムが合格と判定したプログラムのうちの4つは，冗長な命令が含まれており，用意した正解プログラムと全く同じでないにも関わらず認識されている．このことから，本システムのプログラム認識のアプローチが有効であることもわかる．

学生の様子を観察していると，本システムで採用した演習用ダイアログウィンドウを用いた合否判定のアプローチは学生に受け入れ易かったようである．ただし，学生に書いてもらった感想では，システムの動作不良や使い勝手に関して20項目以上の指摘を受けており，これらについて本質的なものは改善する必要がある．

6 むすび

提示した課題に対して学生が作成したプログラムの合否判定を支援するシステムの実現方法を提案した．実験結果より，現在のシステムの使い勝手等を修正することにより実用が可能なこと，教員の合否判定タスクの負荷を大幅に減少できることの見通しを得た．本システムは，我々が行っているような授業形態のみならず，宿題としてプログラムを作成させる場合のプログラム提出用のシステムや遠隔授業における合否判定支援ツールとして用いることも可能であり，幅広く役立つものと考えられる．

今後の課題として，まず，今回の実験で行った形式のゼミを数回行い，システムの洗練化を図った後，実際の授業で使用して評価を行いたい．さらに，問題情報，照合知識の管理ツールやログの解析ツール，解答例のCASLプログラムから一般化表現を自動生成するツールなどについて検討を行いたい．

謝辞 本研究に関してご助言を頂きました奥田健三先生に感謝いたします．

参考文献

[1] Adam,A. and Laurent,J.P. :LAURA, A System to Debug Student Programs, Artificial Intelligence, Vol.15, pp.75 – 122, 1980.

[2] Johnson,W.L. : Understanding and Debugging Novice Programs, Artificial Intelligence, Vol.41, pp.51 – 97, 1990.

[3] 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.

[4] 海尻賢二 : 2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, 信学論, Vol.J78-DII, No.2, pp.321 – 332, 1995.

[5] 渡辺博芳 : WCASL, CASL & COMET Simulator for Windows, http://www.ics.teikyo-u.ac.jp/wcasl/.

[6] 武井惠雄, 筒井多圭志, 荒井正之, 渡辺博芳 : 情報技術(IT)教育の教育基盤, 平成8年度情報処理研究集会, 文部省・名古屋大学共催, 1996.