

CPUとアセンブラ授業のための合否判定支援システムにおけるプログラム認識の改善

5S-1

渡辺 博芳

荒井 正之

武井 恵雄

帝京大学理工学部

1. はじめに

我々は、CASLとCOMETを対象として、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の判定を支援するシステムを試作し、既に発表した[1]。本システムは、学生のプログラムの動作評価と解答例プログラムとの照合によるプログラム認識を行うことで、合否判定を支援する。プログラム認識処理は、学生のプログラムが出題の題意を満たすような実現方法か、教員の予想する実現方法かを検査するのが主な目的とする。ところで、現行のシステムでは学生のプログラムに冗長な命令が含まれる場合でも、動作が正しく、解答例と同じ実現方法であれば、合格と判定する。しかし、冗長な命令はプログラムの処理効率や可読性に影響を与えるので、それらが不要でないことを指摘し、場合によっては削除させるのが望ましい。本稿では、プログラム認識処理における冗長なプログラムに対する判定に関する現システムの改善法について述べる。

2. 合否判定支援システムの概要

(1) システム構成 本システムは、サーバとクライアントから構成される。学生が使用するクライアントパソコン上には、CASLとCOMETのシミュレータWCASL[2]に合否判定機能を追加したシステムを実現した。一方、サーバは合否判定のための問題情報、照合知識をクライアントの要求に応じて提供したり、合否判定に関するログを記録する。

(2) 合否判定処理の流れ 合否判定処理の流れを図1に示す。合否判定要求があると、最初にプログラムの動作評価を行う。プログラムの動作はあらかじめ用意したテストデータに対してプログラムを実行することで行う。プログラムの動作が正しくない場合は不合格となり、正しい場合はプログラム認識に進む。プログラム認識はプログラムの照合と冗長な命令の処理

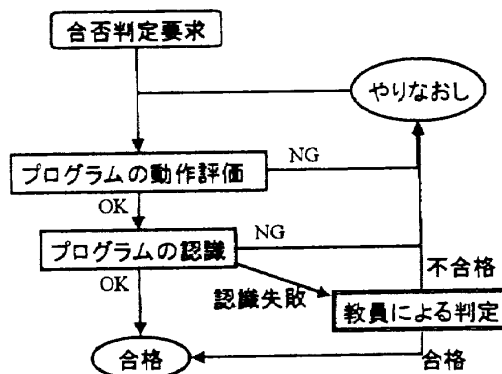


図 1: 合否判定処理の流れ

から成る。まず、あらかじめ用意した解答例プログラムとの照合を行い、照合に失敗した場合は、人間である教員が合否を判定する。プログラムの照合に成功した場合は3.で述べる冗長な命令の処理を行う。これが今回の改善点である。

(3) プログラム照合処理 学生のプログラムを複数の解答例プログラムと照合をとる。同じ実現方法に従う場合でも、僅かな違いによりプログラムに多数のバリエーションが生じるため、解答例プログラムの記述や照合によって、(a)使用するレジスタ番号の違い、(b)使用するラベルの名称の違い、(c)使用する命令の違い、(d)実行結果に影響を及ぼさない程度の命令の実行順序の違いなどを吸収しなければ、蓄えるべき解答例プログラム数が膨大になる。そこで、解答例プログラムでは汎用レジスタを用途ごとに通し番号で表現する。また、何通りかの方法で実現できる命令列に関しては、一般化表現として新しい命令コードを与えて定義する。例えば、レジスタに値を設定する操作は、LEA命令でも、LD命令でも実現できるが、その操作をSETという新しい一般化表現コードで表現する。新しい一般化表現コードは照合知識を使ってCASL命令に対応付ける。これにより、(c)の違いが吸収される。照合処理では、解答例と認識対象プログラム間のレジスタの対応とラベルの対応が無矛盾になるような命

```

PRG    START
      LEA    GR2,0    ;冗長
      LEA    GR1,0
      SUB    GR1,AA
      CPA    GR1,CO    ;冗長
      JPZ    SKIP
      LD     GR1,AA
SKIP   ST     GR1,BB
      EXIT
AA     DC    5
BB     DS    1
CO     DC    0
      END

```

図2: 冗長な命令の例

命令の対応付けを行う。このとき、命令の出現する順序についての制約は設けない。既に動作評価で実行結果を確認しているため、順序が違った場合でも実行結果に影響を及ぼさない程度の違いと判断できるためである。このような処理によって、(a)(b)(d)の違いも吸収でき、比較的少数の解答例を用意すればよい。

3. 冗長な命令に対する処理

(1) 冗長な命令 ある命令、あるいは命令列を削除しても、そのプログラムの実行結果が変わらないとき、その命令、あるいは命令列は冗長であると呼ぶことにする。前述のように現システム[1]では冗長な命令を含む場合も合格と判定するが、冗長な命令であることを指摘することが重要である。図2にAAの数値の絶対値をBBに保存するプログラムにおける冗長な命令の例を示す。第2行のLEA命令はそれ以後でGR2が全く用いられないので、冗長である。また、第5行のCPA命令も冗長である。第4行のSUB命令でフラグレジスタが適切に設定されるからである。合否判定の観点からは、冗長な命令を削除しなければ合格としないケースと冗長であることを指摘するだけで合格とするケースがあると考えられる。第2行のような命令は前者で、第5行のような命令は後者とするような方針もあろう。そこで、合否判定支援システムとしてはそれらが扱える枠組みが必要である。

(2) 冗長な命令の判定処理 冗長な命令を削除させるか、警告に留めるかの判定、及びそのとき提示するメッセージの内容は個々のケースによって異なると考えられる。そこで、あらかじめ定義した冗長な命令の判定知識を用いて判定することを基本方針とする。一方で冗長な命令の全てのケースを網羅することは困難であるので、判定知識が適用できない場合は、デフォルトの処理を行うこととする。冗長な命令の

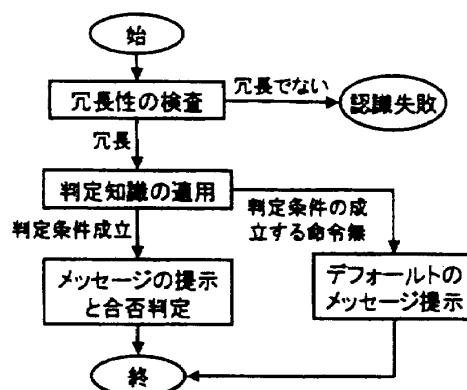


図3: 冗長な命令の判定処理の流れ

判定知識は、(a)判定条件(手続き)、(b)合格判定における削除の必要性(論理値)、(c)提示するメッセージ(文字列)の情報から構成する。(a)の判定条件は、例えば、「レジスタに値を設定する命令の後、そのレジスタが全く用いられていない」などを示す。プログラムリストに対するこのような検査は手続き的な処理となるので、判定条件は手続き的に記述してクライアントシステムに組み込む。(a)の条件が成立する場合、(b)が真であれば不合格となる。また、合否に関わらず、条件が成立した場合は(c)のメッセージを提示する。(b)(c)については教員が設定可能にする。

冗長な命令の候補は、プログラム照合において解答例の命令に対応付けられない命令であるので、それらの各命令について図3の処理を行う。冗長性の検査ではその命令を削除したプログラムリストの動作評価を行い、正しく動作すれば冗長であると判定する。命令の削除の際にはラベルを考慮した処理が必要となる。次にあらかじめ定義された判定知識を適用し、判定条件の成立する知識が存在すれば、その知識に定義されたメッセージを提示し、(b)の情報に基づき、合否を判定する。冗長でない命令が存在する場合は実現方法が異なると考え、プログラム認識は失敗となる。

4. おわりに

本稿で述べた冗長な命令の扱いに関する処理を現在実装中である。今後は、今回の改善法による効果の評価を含めて、実際の授業で使用してシステム全体の評価を行う予定である。

参考文献

- [1] 渡辺博芳, 荒井正之, 武井恵雄: CPUとアセンブラ授業のための合否判定支援システム, 情処研報, Vol.98-CE-48, pp.61-68, 1998.
- [2] 渡辺博芳: WCASL, CASL & COMET Simulator for Windows, <http://www.ics.teikyo-u.ac.jp/wcasl/>.