3Q－5　プログラムにおける因果理解の表現と空欄補充問題への応用

曽我真人（和歌山大学），柏原昭博，豊田順一（大阪大学産業科学研究所）

１．はじめに

与えられた仕様を満たすプログラムを設計，記述する能力とならんで，与えられたプログラムを理解する能力は重要である．プログラムを理解する能力は，過去のプログラムを再利用したり，複数の人間で共同開発を行う場面で，他人の記述したプログラムを理解する時などに必要とされる．筆者らは，この理解能力を向上させる効果的な一つの手法として，適切なプログラム空欄補充問題を学習者に与えることを考えている．

空欄を補充するためには，前後のステートメント間の因果関係を推論する必要があり，文脈を理解しなければならない．その因果関係を推論する能力を養うためには，プログラム中の数カ所を空欄にした空欄補充問題を学習者に与えればよい．理解能力の向上を効果的に行うためには，与える問題の空欄補充の箇所を適切に設定する必要がある．設定箇所は，学習者にプログラムの何を理解させたいか，学習者にどれだけの推論負荷をかけたいか，などの要素により決定されると考えている．学習者の能力に応じた推論負荷をかけることにより，効果的な学習が期待できる．それらのことを検討するためには，プログラムの理解の表現が必要である．

本稿では，プログラムの理解の表現を考察し，それをもとに学習者の理解能力を効果的に向上させるような空欄補充問題は，どのように作成すればよいかについてＣ言語で作成されたプログラムを例に論ずる．

２．プログラムの理解の表現

ここでは，空欄補充問題の設計者の立場から，プログラムの理解を完了した状態を想定して，理解の表現を考える．そして，プログラムの理解を，a) ブロックのまとめあげ，b) データフローのトレース　の２つの大きな視点からとらえる．

a) については，さらに，振る舞い，構造，機能の３つの視点からとらえる．プログラムはそれぞれのステートメントが何らかの動作を行うので，これを振る舞いととらえる．そして，いくつかのステートメントが集まって，まとまった振る舞いをおこなう単位があり，このまとまりをブロックととらえる．ブロックはより大きなブロックの部品となり，文やブロックをノードとして階層構造を形成する．そして，振る舞いはノードの属性として表される．図１は正の整数を入力させ，階乗を計算し表示するプログラム例の理解の表現図である．機能は，各ブロックの振る舞いがプログラム中で果たす役割をいう．より具体的には，振る舞いがブロックの中身の動作を指すのに対して，機能はブロックの入出力をいう．機能もノードの属性として表現される．

プログラム空欄補充問題では，通常，全体の機能は問題文の中で与えられる．入出力例として機能が示されることもある．場合によっては，ブロックの機能も与えられることがあるが，どのブロックの機能であるかまでは明記されないのが普通である．

b) については，複数のステートメントが共通の変数を使用している場合，ステートメント間にデータの流れによる因果関係が生じる．１つのブロック内でもデータフローは存在するが，異なるブロック間でデータフローが存在する場合には，ブロック間の因果関係を表すことになる．

このような理解の表現から，学習者の理解は，プログラムを読んで振る舞いをシミュレートして，ボトムアップ的にブロックをまとめあげてゆくと同時に，ブロックの振る舞いを抽象化して機能を理解すること，もしくは，問題文に与えられた機能をブロックに対応づけ，それと同時にデータフローをとらえ，ステートメント間やブロック間の因果関係を明らかにすること，としてとらえることができる．

３．空欄補充問題の設計指針

空欄の設置個所によって，空欄補充の際に学習者にかかる推論負荷の大きさを制御することにより，学習者の能力に応じた難易度の空欄補充問題を作成することができる．その際に，ランダムに空欄の箇所を選んで比較しても難易度の順序づけをすることは困難であるから，負荷制御を行うことができない．そこで，筆者らは，効果的な空欄補充問題を作成するために，次のような指針を考えている．第一段階は設計者が何を意図してその問題を学習者に与えるかにより，プログラムのある箇所を空欄として設定する．これを主空欄と呼ぶ．第二段階では，第一段階で設定した主空欄補充の難易度を上げるために新たな空欄を設定する．これを副空欄と呼ぶ．すなわち，副空欄の設置個所や設置数によって，主空欄補充の難易度を制御し，学習者にかかる推論負荷を制御しようとするものである．

第一段階では，プログラムの特定の部分を重点的に理

A Representation of Causal Understanding in Programs and Application to Blank-filling Problems
Masato SOGA , Faculty of Economics, Wakayama Univ. 930 Sakaedani, Wkayama, 640 Japan
Tel: 0734-54-0361 Ext. 3324,　Fax: 0734-54-0390,　E-mail: soga@eco.wakayama-u.ac.jp

解させたい部分を主空欄とする．

第二段階では，副空欄が主空欄に対してどのような関係にあるかにより，難易度が異なる．難易度に関係すると考えられるファクタには次の3つが考えられる．

1)副空欄の存在する場所が，主空欄と同一ブロック内か，異なるブロック内か．

2)副空欄の存在する場所が，主空欄に隣接しているか否か．

3)副空欄に補充されるべきものが，主空欄と同一変数を含むか否か．

1)については，同一ブロック内の方が異なるブロック内よりも難易度が高くなると考えられる．空欄を補充するためには，そのブロック内の各処理を理解し，まとめあげて，ブロック全体の振る舞いを理解することが必要であるが，ブロック内の空欄の数が増えるほど，手がかりが減り，まとめ上げが難しくなると考えられる．

2)については，隣接している方がそうでない場合よりも難易度が高くなると考えられる．これは主空欄の前後の因果関係を推論する手がかりが減少することと，空欄が連続すると，それらを埋めるのは小規模ながら設計タスクとなるからである．

3)については，同一変数を含む方が難易度が高くなると考えられる．空欄を補充するためには，補充される変数の意味を理解すると同時に，その変数が使用されている部分との因果関係を推論する必要があるが，その手がかりが減るからである．

1）2）は，学習者が，ブロック内の振る舞いをまとめあげて機能を理解しようとする際にかかる負荷の大きさに関係している．3）はブロック間の関係を理解しようとするときの負荷の大きさに関係している．

例えば，図1の例の場合，主空欄をプログラムの11行目（理解表現の斜線部）とし，副空欄の候補として7行目，12行目，16行目（いずれも理解表現の着色部）を比較してみる．12行目は前述の1）～3）をすべて満たしている．それに対して，7行目は3）を満たすのみである．また，16行目はどれも満たさない．したがって，7行目を空欄にする方が16行目を空欄にするよりも主空欄補充に負荷がかかり，さらに，12行目を空欄にした方が7行目を空欄にするよりも主空欄補充に負荷がかかり，難易度が増す．

おわりに

本稿では，プログラムの理解の表現と，それを利用して，プログラム空欄補充問題の空欄設置個所の違いによる学習者にかかる負荷の大きさについて述べた．今後，実験を通じて検証して行く予定である．

```
1  #include <stdio.h>

2  main()
3  {
4     int i, j;
5     long l = 1;

6     printf("正の整数を入力して下さい：");
7     scanf("%d", &i);
8     j = i;
9     if( i > 0 ){
10       while( i > 0 ){
11          l *= i;
12          i--;
13       }
14       printf("%dの階乗は%ld¥n", j, l);
15    }else{
16       printf("入力ミス¥n");
17    }
18 }
```

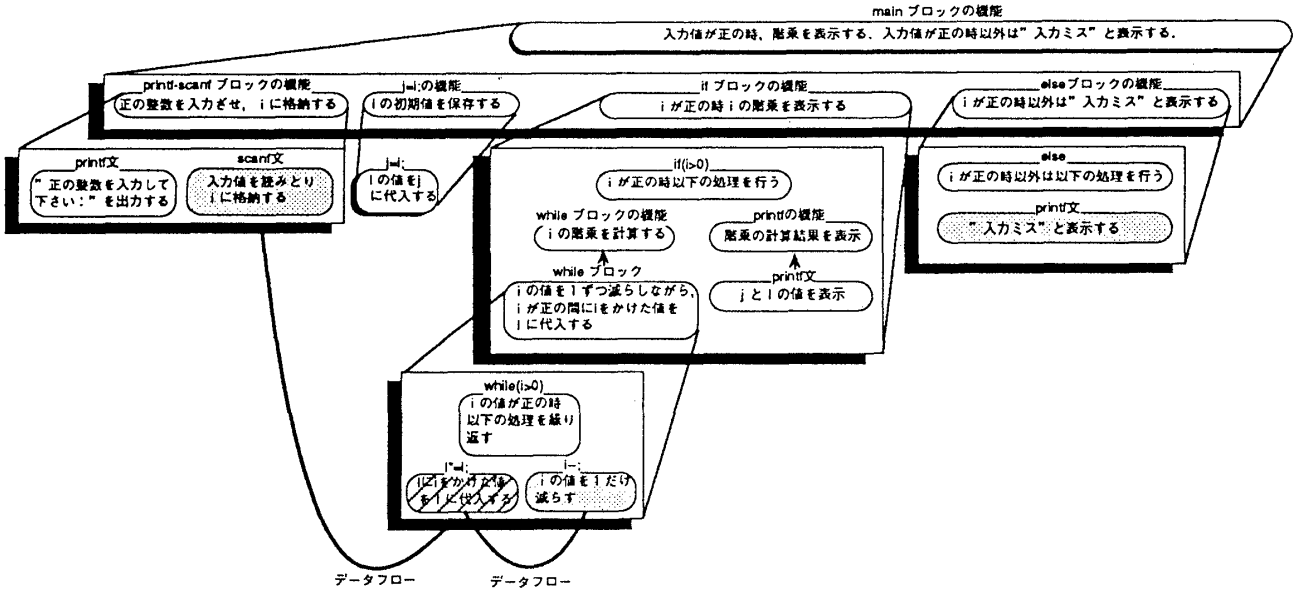

図1　プログラム理解の表現例