

ソフト開発用データフローチャートとそのコーディング手順について(2)

3M-11

相浦 利治 , 大原 茂之 , 小高 明夫
東海大学

1. はじめに

DF型TSチャート¹⁾²⁾は、モジュールを関数としてC言語のプログラムを作成することが可能である。また、モジュールの駆動をTSチャートによってコントロールし、記号間のデータの受渡しをアークによって意味付けることができる。ここでは、8種類のDF型TSチャート記号を定義し、その範囲で記述されたチャートをC言語のプログラムに自動変換するためのアルゴリズムを述べる。

2. 本文

2.1 DF型TSチャート記号について

ここでは、次のDF型TSチャート記号とその働きについて述べる。

《処理記号》 (1)モジュール記号 (2)マクロモジュール記号 (3)モジュール制御モジュール記号 (4)排他的処理記号 (5)外部変数記号

《接続記号》 (6)入出力アーク記号 (7)リードアーク記号 (8)シーケンスアーク記号

【定義1】図1(1)は、モジュール制御モジュール記号である。図1(4)のようにモジュール制御モジュールを接続した場合、条件Qが成立する間、sを駆動する。

【定義2】図1(2)は、排他的処理記号である。図1(5)のように排他的処理記号を接続したとき、 $y_1 \sim y_n$ の中で出力可能な任意のデータの1つをxに出力する。

【定義3】図1(3)を外部変数記号という。外部変数記号は、入力データと同じデータを出力する。

2.2 DF型TSチャート記号とC言語の関係

【定義4】SをDF型TSチャート記号の処理記号の集合とする。Aをアークの集合とすると $S \times A \times S$ の部分集合をDF型TSチャートの接続集合と呼ぶ。また、 $(a, b, c) \in S \times A \times S$ なる元をaからcへの接続と呼ぶ。

接続が (x, k, y) であるDF型TSチャートをC言語で記述する2つの方法を次に述べる。第1の方法は、yの中にxを埋め込んで記述する方法である。これを「埋

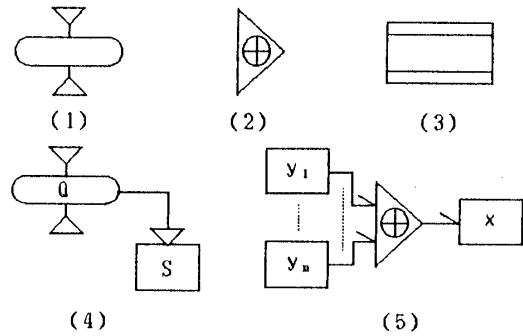


図1 DF型TSチャート記号

め込み法」と呼ぶ。第2の方法は、まずxとyの間に仮想的に変数を用意する。この変数を仮想変数と呼ぶ。xは仮想変数にデータを出力する。そして、yが仮想変数の値を入力する方法である。これを「仮想変数法」と呼ぶ。「埋め込み法」は、出来上がったプログラムの入れ子構造が深くなり、理解しにくくなってしまふ。また、xの出力先がn個存在した場合、実行時間はxがn回駆動するためオーバーヘッドが大きくなる。以上の理由により、C言語への記述法は「仮想変数法」を用いる。〔証明〕「仮想変数法」を用いることが、可能であることを証明する。

$\exists x \exists y \in S$ において、xにiを入力するとjを出力する。yにjを入力するとkを出力する。 (x, a, y) にiを入力した場合kが出力される。このときに、xとyの間に仮想変数zを用いたとする。 (x, a, z) 、 (z, b, y) においてxにiを入力したとき、yからkが出力されることは明かである。

【定義5】 $\exists x \in S$ に対し、 $\{y \mid (x, a, y) \in \text{DF型TSチャートの接続集合}\}$ なる、xからyまでをxの出力の有効範囲とよぶ。

[補題1] $\#(\{(x, a, y)\})=1$ なるyに対し、 $\{z \mid (y, b, z) \in \text{DF型TSチャートの接続集合}\}$ とする。yは、xからの入力をyの有効範囲に受渡した後に、xが再びデータを出力するまで出力不可能である。これによりyは、出力可能、不可能を示す状態を持つと考えられる。

<規則1>図2(1)に示す構造体が、仮想変数の構造体

```

struct <データ型>vv{      for(x=0;x<5;++x){
  <データ型> z;          f();
  int f;                  }
}
(1)                      (2)

xor(z1,z2,&z3);
(3)

```

図2

である。これを仮想構造体と呼ぶ。zは、受渡しを行うデータの内容を代入する。fは、zに値が代入された時点で1を代入する。そして、有効範囲の出力が終了した後に0を代入する。

<規則2> 図1(2)においてQをfor(x=0;x<5;++x),sをf()とした場合は、図2(2)に示すC言語で記述する。

<規則3> 図1(4)においてn=2であり、y1,y2の出力は仮想構造体z1,z2に代入し、排他的処理記号の出力が仮想構造体z3に代入されるとする。この場合排他的処理記号は、図2(3)に示すC言語の関数となる。排他的処理記号の関数の引数は、記号へ接続された入力を上から順に記述していく。排他的処理記号関数の出力は、引数の左から入力可能なものを構造体のfから検索し、その仮想変数の内容を出力する。

2.3 フィードバックループ

【定義6】 接続(z0, a1, z1), (z1, a2, z2), ..., (zn-1, an, zn)において、z0=znならばznの出力をz0にフィードバックするという。

[補題2] 図3(1)における排他的処理記号を用いたフィードバックループは、次のような動作となる。外部変数hには値が入力されているものとする。s1の実行後s2にデータが出力される。s3は、出力可能なhを読み込みs2へ出力し、s3に入力されているすべての仮想構造体のfを0にする。s2は入力データがそろった時点で、実行を開始しデータを出力する。再びs1が実行されs2にデータが出力される。s3は、出力可能なs2の仮想変数を読み込み、出力する。s1をf1,s2をf2,hをxとした場合、C言語では図3(2)のように記述される。

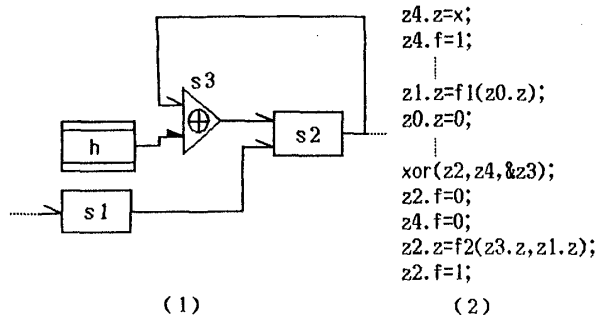


図3 フィードバックループ

2.4 記号の接続とコーディング順序の関係

【定義7】 接続(s, a1, t), (t, a2, u)において、(r, a0, s)なるa0が存在しない場合、sをコーディング可能記号と呼ぶ。

【定義8】 記号のコーディング順序を示す番号をコーディング番号と呼ぶ。コーディング番号は、コーディング可能記号に付けられる。

[補題3] $\exists y \in s$ に対し、 $\{x \mid (x, a, y) \in DF \text{型TSチャートの接続集合}\}$ において、すべてのxにコーディング番号が付けられている、もしくは出力をyにフィードバックするならば、yはコーディング可能記号である。

2.5 コーディングアルゴリズム

以上の定義より、DF型TSチャートからC言語へのコーディングアルゴリズムを述べる。

(A1) マクロ展開: 階層0上に存在する全てのマクロモジュールをマクロ展開する。

(A2) コーディング番号付け: 【定義7】、【定義8】に基づいて、全てのDF型TSチャート記号の処理記号にコーディング番号をつける。

(A3) コーディング: DF型TSチャート記号の処理記号につけられたコーディング番号順にコーディングを行なう。この際、入出力アーク、リードアークによって接続された記号間には、仮想構造体を用意する。

3. おわりに

本報告では、TSチャートによるデータフローのコントロールと、フィードバックループ記述を含む自動コーディングのアルゴリズムを示した。この他に副作用アークなどを含むDF型TSチャートのコーディングアルゴリズムを現在検討中である。これらについては、次の機会に報告したい。

謝辞 本研究を進めるに当たり、多くの面でご援助いただいた、本学工学部長萩三二教授、電子工学科主任飯田昌盛教授に感謝の意を表します。

参考文献

- 1) 大原茂之: 木構造化チャートによるデータフローの制御について、第36回情報処大全3L-1, 1988
- 2) 相浦, 大原, 小高: ソフト開発用データフローチャートとそのコーディング手順について、第36回情報処大全3L-2, 1988
- 3) 大原茂之: 木構造化チャートによる階層構造的プログラミング、東海大学紀要工学部、Vol. 27, No. 1 1987
- 4) 大原茂之: 木構造化チャートによるプログラム開発・保守技法、情報処理学会論文誌、Vol. 27 No. 10, 1986