4S-2

モジュールの予見性を評価する一方法

チュー・ミンフォン　宇都宮 公訓　藤原 譲
(筑波大学)

1．はじめに

複合設計ではモジュール独立度が高い設計ほどよいプログラム構造を持つ設計と考えており、共通結合や外部結合はモジュール独立度を低くする要因とみなし、これらを極力排除するような設計のガイドラインを設定している[1]。共通結合や外部結合はモジュール間で共有する状態を作ってしまい、モジュール間の独立度を低くするだけでなく、モジュールの予見性をなくしてしまう。モジュールの予見性とはモジュールの出力がモジュールの入力だけで決まる性質のことである。本研究は、多くのモジュールからなるFORTRAN プログラムについて、モジュール間データ受渡しまで考慮して、モジュールの予見性を評価する一つの方法を提案している。

2．プログラムのモデル化

複数のモジュールからなるFORTRANプログラムを図1のような状態機械でモデル化する。Mi(i=1,2,..)は状態機械であり、モジュールに対応している。各々の状態機械は固有の状態LSi(i=1、2、..)すなわち個別状態を持つと共に、共通状態GSを共有している。LSiは個々のモジュールの静的な局所変数に対応し、GS は COMMON ブロックに対応している。

一般に、Mi は引数を介してモジュール間でやり取りする入出力と、プログラムの外部に対応する入出力を行なう。前者による入力を引数入力 AIi、出力を引数出力 AOi 、後者による入力を外界入力 PIi 、出力を外界出力POi、と呼ぶことにする。次のような関数の組でMi の定義をすることができる。

POi = FPOi(AIi, PIi, GS, LSi, CMi)
AOi = FAOi(AIi, PIi, GS, LSi, CMi)
LSi = FLSi(AIi, PIi, GS, LSi, CMi)
GS = FGS(AIi, PIi, GS, LSi, CMi)

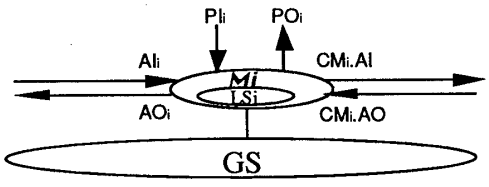

図1: FORTRANプログラムの状態機械モデル化

ただし、CMiはMiが直接呼び出す状態機械を通して与えられる効果である。現実のプログラムでは例えばAOiがAIi、PIi、GS、LSi、CMiの可能なすべてのデータの関数になっているわけでなく、それぞれの部分集合の関数である。AOiなどの状態への従属性を議論するために、例えばGS[FAOi]と書いて、「Miに対応するモジュールのソースコードだけから判断して、AOiを与える関数FAOiにおいて、実際に独立変数になっていると見なせるGSの部分集合」を表し、¦GS[FAOi]¦でその部分集合の濃度を表すものとする。

AOiが状態独立であり、AOiがAIiとPIiだけから決まるとき、「Miは（AOiに関して）予見性がある」[1]とか、「Miはカーテジアン」[2]という。次のいずれかの条件を満足するMiはAOiに関して予見性がある。

(PR1)
¦LSi[FAOi]¦ = 0 & ¦GS[FAOi]¦ = 0 & ¦CMi[FAOi]¦ = 0

(PR2)
¦LSi[FAOi]¦ = 0 & ¦GS[FAOi]¦ = 0 & ¦CMi[FAOi]¦ > 0
& PR[CMi*]

ただし、CMi*は、CMi を含めて、それより先で呼び出されるすべてのモジュールを表し、PR[CMi*]はCMi*が予見性があることを表している。CMi*をMiの下流モジュールとよぶ。

CMi*のいずれかのモジュールでGSを変化させると、一般に、Miの予見性はなくなる。しかし、次の場合は予見性が保存される。

(PR3)
¦LSi[FAOi]¦ = 0 & ¦GS[FAOi]¦ = 0 & ¦CMi[FAOi]¦ > 0
& (Aj(¦LSj[FAOj]¦ = 0 & ¦GS[FAOj]¦ = 0))

ただし、Ajは、「CMi*に属するすべてのモジュールに対して」の意味で使っている。

3．モジュールの予見性を評価する方法

FORTRANのプログラムについては、Miの予見性の評価に際して、LSi、GS、PI、POは容易に識別できる。しかし、AOとAIの識別は簡単ではない。次のような手順でAOとAIを決定する。

A Method to Evaluate the Predictability of Modules

Choo, Min Fong , Kiminori Utsunomiya , Yuzuru Fujiwara
University of Tsukuba

[引数入出力決定手順]

（1） 各モジュールごとに、引数を個々に調べ、そのモジュールにおけるR/W値を決定する。ただし、モジュールR/W値は次のようにして決める。

n ：読み出しも書き込みもしていない
r ：読み出しだけをしている
w ：書き込みだけをしている
rw ：読み出しも書き込みもしている
nu ：読み出しも書き込みもしていないが、下流モジュール呼び出しの引数になっている
ru ：読み出しだけを行ない、下流モジュール呼び出しの引数にもなっている
wu ：書き込みだけを行ない、下流モジュール呼び出しの引数にもなっている

（2） （1）で値n、r、w、rwのいずれかをとる引数については、その値を実効R/W値とする。（1）で値nu、ru、wuのいずれかをとる引数については、（3）の操作を行なって実効R/W値を決定する。

（3） 直接呼び出しているモジュールにおけるその引数のモジュールR/Wとの間で、カスケード演算［表1］を次々と施していく。同じ引数が、あるモジュール内で二つ以上の直接下流モジュールの呼び出しの引数になっているときは、上記の計算後、重ね演算［表2］を施す。

（4） 実効値R/Wを用いて、AO、AIは次のように決定する。

rであればAIである
wであればAOである
rwであればAIでもAOでもある
nであればAIでもAOでもない

モジュールMiのAOi、AIiの個々の引数とGSやLSiとの関係も、引数入出力決定手順とほとんど同じ手順で決定できる。ここではGSとの関係だけについて記述するが、LSiとの関係も全く同様にして決定できる。

[引数対GS関係識別手順]

（1）モジュールMi内でのその引数のモジュールR/Wの値を決定する。ただし、モジュールR/W値は次のようにして決める。

n ：GSがこの引数に従属してなく、かつ、この引数もGSに従属していない
r ：GSはこの引数に従属しているが、この引数はGSに従属していない
w ：この引数はGSに従属しているが、GSはこの引数に従属していない
rw：GSはこの引数に従属しており、かつ、この引数もGSに従属している
nu：Mi内ではnで、この引数に従属する変数が下流モジュール呼び出しの引数に使われている
ru ：Mi内ではrで、この引数に従属する変数が下流モジュール呼び出しの引数に使われている
wu：Mi内ではwで、この引数に従属する変数が下流モジュール呼び出しの引数に使われている

ただし、nu、ru、wuの値とする場合、下流モジュール呼び出しの引数で値を受け取る引数は同じ名前でなければならない。

（2）と（3） 引数入出力決定手順の（2）、（3）と全く同じである。

（4）（2）、（3）で決まった実効値R/W値を用いて、GSとの関係を次のように決定する。

n ：この引数はGSと関係していない
r ：GSはこの引数に従属しているが、この引数はGSに従属していない
w ：この引数はGSに従属しているが、GSはこの引数に従属していない
rw：GSはこの引数に従属しており、かつ、この引数もGSに従属している

第1オペランドの値	第2オペランドの値						
	n	r	w	rw	nu	ru	wu
nu	n	r	w	rw	nu	ru	wu
ru	r	r	rw	rw	ru	ru	rw
wu	w	rw	w	rw	wu	rw	wu

表1 カスケード演算の定義表

#1＼#2	n	r	w	rw
n	n	r	w	rw
r	r	r	rw	rw
w	w	rw	w	rw
rw	rw	rw	rw	rw

表2 重ね演算の定義表

4. おわりに

今後、これらの手順を利用したFORTRANプログラム理解支援ツールを開発し、保守支援ツール体系に組み込んでいく。

参考文献

[1] 國友義久他(訳): ソフトウェアの複合/構造化設計、近代科学社

[2]Britcher,R.N. and J.J.Craig: Using Modern Design Practices to Upgrade Aging Software Systems, IEEE Software,May1986, pp.16-24