良い『メモリ図』の生成

6R-9

川副 博　徳山 豪　岩野 和生

日本アイ・ビー・エム株式会社 東京基礎研究所

1. はじめに

ソフトウェア・ライフサイクルの中で，保守は大きな問題である．他人が作成したプログラム，作成されてから年月が経たプログラムなどを保守するためには，プログラムを理解することが必要である．プログラムを理解するには，入力，出力，途中段階のデータの構造を知る必要がある．

アセンブリ言語では，メモリ上に，自由に変数（領域）をとれる．レコード型を実現するために，同一のアドレスに二つ以上の変数を割当てる（アリアスと呼ぶ）こともある．このようなプログラムを理解するには，変数（領域）の割当てを理解する必要がある．メモリ参照速度をあげるため，アセンブラが自動的に変数のアドレスをワード境界に合わせることがある．このためソース・プログラムには無い（変数と変数の間の）隙間ができ，プログラム理解の妨げになる．

我々はアセンブリ・プログラムのメモリ上の変数の割当てを表す図（『メモリ図』と呼ぶ）［1］を自動的に作成するツールを試作した．以下では，メモリ図について述べ，メモリ図を作る際に出会った問題をアルゴリズムの見地から見直して述べる．

2. メモリ図

メモリ図とは，アセンブリ・プログラムのなかの変数（領域）宣言部分を図示化したものである．

［図1］は変数宣言部分の例，［図2］が対応するメモリ図である．メモリ図では，個々の変数は開始アドレスから終了アドレスまでの区間を横辺とする太さ1の帯として表される．同一アドレスを含む二つの変数はメモリ図では，異なった段に配置されることにより，重なることなく示される．画面／紙面の行数には制限があるので，メモリ図を構成する時には，段数をあまり増やさない必要がある．

このツールを用いて実用に供されているプログラム(例：O.S.のｽｹｼﾞｭｰﾗ，ｴﾃﾞｨﾀ)のメモリ図を作成した．

3. アリアスを考慮したメモリ図の実現の試み

アリアスを含む変数宣言は，（アリアスがあるアドレスでの帯の置き方に複数の方法があるので）メモリ図は複数種類ある．アリアスを使う目的の一つにレコード型の実現がある．レコード型を実現するためには，レコード型の各下位フィールドの長さの和を持つ変数を宣言し，その変数を下位フィールド毎に区切って，変数宣言する．レコード型を持つ変数宣言に対してメモリ図を作成したとき，レコード型の上位フィールドに対応する帯の下に各下位フィールドに対応する帯を配置すると，データ構造は把握しやすい．試作したツールではレコード型に関する配置問題までは気を配っていない．以下で，そこまで配慮した算法を紹介する．

```
000000      3 MEIBO    DSECT
000000      4 KOJIN    DS      0CL32
000000      5 NUMBER   DS      F
000004      6 NAME     DS      0CL16
000004      7 SEI      DS      CL8
00000C      8 MEI      DS      CL8
000018      9 OTHER    DS      D
000020     10          ORG     OTHER
000018     11 BIRTH    DS      0CL6
000018     12 YEAR     DS      CL2
00001A     13 MONTH    DS      CL2
00001C     14 DATE     DS      CL2
00001E     15          ORG     OTHER
000018     16 SIZE     DS      0CL6
000018     17 BUST     DS      H
00001A     18 WAIST    DS      H
00001C     19 HIP      DS      H
           20          END
```

図 1. アセンブリ・プログラムの変数宣言の例

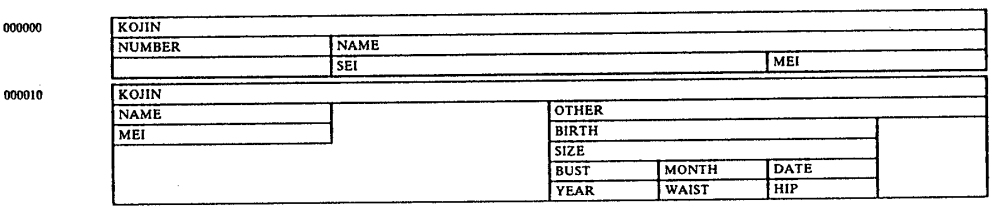

図 2. メモリ図の例

Generating Good Data Structure Chart

Hiroshi KAWAZOE, Takeshi TOKUYAMA and Kazuo IWANO
Tokyo Research Laboratory, IBM Japan, Ltd.

4. メモリ図の構成の定式化

アセンブラの一つの変数を開始アドレスから終了アドレスまでの数直線上の区間とみなすと，変数全体は区間の集合だとみなせる．今，交差しない区間の集合を無交差集合と呼ぼう．メモリ図で一段に配置される変数らは（同一アドレスを含まないので）無交差集合に対応している．従って，できるだけ少ない段数に変数らを配置する問題は次のように言い換えられる．
S={I(1),I(2), ... ,I(n)} を区間の集合とする．

問題 1．（最小無交差分割）

Sを無交差集合らに分割し，分割数を最小にせよ．

更に，第3章で述べたように，レコード型の上下関係を反映させた上で，段数最小分割を求めたい．上から数えてi段目に配置される変数たちの集合をAiと書こう．

問題 2．（制約付き最小無交差分割）

Sを無交差集合A1,A2, ... ,Adに分割し二つの無交差集合AiとAjにおいて，i<j（AiがAjより上）であれば，Ajに入る任意の区間らはAiに入る区間を含まない（下位フィールドとして持たない）という条件を付ける．更に，分割数dを最小にせよ．

問題2．を解けば，レコード型の上下関係を保ったメモリ図が求まる．

さて，点xを含む区間数が最大になるxを考え，その時の区間数をqと書き，最大重なり数と呼ぶ（［図2］ではq=6）．メモリ図には少なくともq行必要なことは明らかである．問題1．は次の様に解けることが知られている．

命題 1． Sをq個の無交差集合に分割することができ，O(nlog(n))の計算時間で，最適分割が見出せる．

次の章で，問題2．に関して最適に近い解を効率よく求める二つの算法を考える．

5. 制約付き無交差分割問題の算法

5.1 グラフの分割への帰着

Sの区間I(j)(j=1,2,..,n)に対応する{Vj:j=1,2,..,n}を頂点とした複合グラフをG(S)（有向辺と無向辺とがあるグラフ）を次で定義する．
1)I(i)∩I(j)≠φのとき，ViとVjとを無向辺で結ぶ．
2)I(i)⊃I(j)ならば，ViとVj間の辺にViからVjに向きを付け有向辺にする．

さて，G(S)の2頂点ViとVjの間に，長さ2以上の有向道がある場合，ViとVjの間を直接結んでいる辺を消去する．この様に辺を刈り込んで作ったグラフを$G_0(S)$とする．$G_0(S)$の強独立集合 Aとは，頂点の集合で，Aの2点間には辺も有向道もないものを言う．すると，$G_0(S)$を次の算法で分割したものは，制約付き無交差分割に対応している．

算法1． G = $G_0(S)$,
do while （G が完全グラフでない）
1．Gの極大強独立集合Aを求める
2．Aを出力する
3．Aを縮約してGを変形する
end

この算法が見出す制限付き無交差分割の分割数（メモリ図の段数）は，Step 1.のAの取り方に依存し，分割数を少なくするための種々の工夫が考えられる．
この算法の手間は最悪でも$O(n^3)$である．

5.2 分割統治法

ある実数xをとり，開区間 $(-\infty,x)$に含まれるSの区間の集合を$S_l(x)$，xを含むSの区間の集合を$S_0(x)$，(x,∞)に含まれるSの区間の集合を$S_r(x)$とする．このとき，$S_l(x)$，$S_0(x)$，$S_r(x)$に対して，それぞれ制約付き無交差分割を作り，それらを行数最小になるように最適に合併させて，Sの制約付き無交差分割を作ることを考える．次の再帰的算法が考えられる．

算法 2． Cluster(S)
1．$S_l(x)$，$S_r(x)$がそれぞれn/2個以下の区間しか含まないようにxを選ぶ．
2．S = $S_0(x)$ ならば，区間を長い順に並べて終了．
3．Cluster($S_l(x)$),Cluster($S_0(x)$),Cluster($S_r(x)$)を合併する．
この算法は，Sを最悪でもqlog(n/q)行に配置し（qは最大重なり数），多くの場合，最適に近い配置を求める．この算法は全体で$O(nlog(n)+nq^2log(q))$の手間である．

6. まとめ

アセンブリ・プログラム理解のために，メモリ図の自動作成ツールを試作した．更に，レコード型を作るためのアリアスを見易く配置するツールの開発を目標として，問題を定式化し，グラフ理論を用いた解法と，計算幾何学を用いた解法とを述べた．詳しい解法は［2］を見られたい．

参考文献

［1］Data Areas and Control Block Logic-CP, IBM Corporation, LY20-0896.
［2］Algorithms for Generating Data Structure Chart, In preparation.