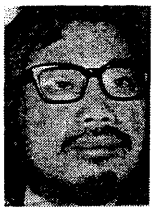
359

"プログラミング画法"の提案



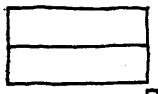
竹内 郁雄
(日本電信電話公社 武蔵野電気通信研究所)

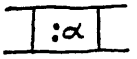
I. もくろみ

データ構造とその処理はなんといっても図に書いて説明してもらうのが一番わかりよい. 本小文は, データ構造とその処理を図式的に表現したものをプログラム言語の中に埋め込めないだろうか?という趣旨のものである.

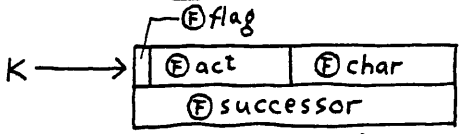
II. とりあえず

以下に述べる文法(画法?)は, 以前私がFORMAL-2という言語の処理系に関係したとき, ドキュメンテーション用に捏造したものである. とりあえずこれをタタキ台として考えてみたい. 話を具体的にするため1語16ビットの機械を想定する.

1.  の如き箱の列はメモリの一部を語単位に分けて表わしたものである. 1語はそれ自身でフィールドである.
2. 1語は  の如くより細かいフィールドに細分化できる. 特に断らぬ限り幅の細い区切りは1ビット, 真中の区切りはバイトの区切りを表わす.
3. ⟶(矢印)は変数またはメモリがその語を指していることを示す. 指し示し方が適当にコード化されているときは $\xrightarrow{f}$ のように書く. ——(棒)はフィールド内に書くべきことを外に書くための補助線である.
4. 変数やメモリ(フィールド)の書き替えは, :(コロン)を添付することによって表わす. 下の対照表は ⇨ の左が平叙文(!)に対応し, 右が書き替え文(命令文)になっている. (平叙文はif-clauseの中では疑問文となる.)

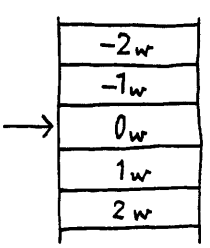
= (等しい)	⇨	:= (代入)	
⟶ (指している)	⇨	⟶: (指すように書き替える)	
 (このフィールドの値はα)	⇨	 (フィールドの値をαに書き替える.)	

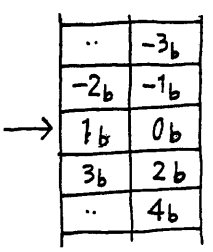
5. いわゆる宣言文は平叙文と見做し, フィールドの定義は次のように行なう.



(Kの指している語のMSBをflag, 下バイトをchar, 残りをact, 次の語をsuccessorと呼ぶ)

6. メモリの内容は[と]でくくって表わす.
7. ある具体的なフィールドを表わすときは, 基本となる語とフィールド名を.(ピリオド)で継ぐ. たとえば
   K.act, [K.act], [K.successor].act など
8. 右のようなフィールドは自明のフィールドとして予め特殊な表し方をする. インデックス修飾に対応するものである. 変数を使って
   $[N.LOCATION_w]$, $C.(IDX+6)_w$
   というような表現も可とする.

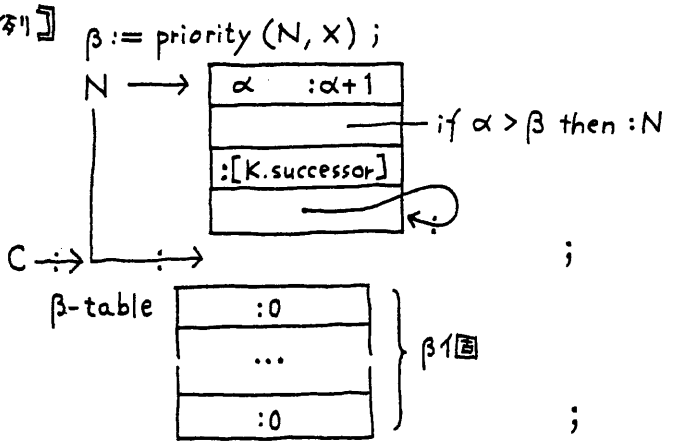




9. | の後はコメント的な補助説明．…は同じようなメモリ（あるいはデータ構造）の列の省略．（要するにこの辺は臨機応変）

10. 特定の語に名前を付けることができる．たとえば，tsukimoto [　　]．

〖例〗



二番目の文は，Nの指している語の内容を1増やし，次の語にはNを書き込み（α>βの場合），以下次々に，[K.successor]と自分自身を指すポインタを書き込み，NおよびCがさらにその次の語を指すように書き替える．
三番目の文はβ-tableを先頭からβ個クリアする．

ここで，平叙文に対応するものはデータ構造に対する処理が行なわれる前の状態を表わし，：による書き替えは処理後の状態を表わしている．つまり一つの図の中に前後する二つの時間の状態が記述されており，かつその間の変化は並行して同時に起ったと考えるわけである．このように実行の順序が本質的でない複数の処理が並列してうまく書けることに注意しよう．実際，FORMAL-2のドキュメンテーションに使用した経験からいって，このような画法をたとえばBLISS（この場合，＜と[]の意味を逆にしなければならない）に埋め込むと大変効果的である．厳密性について問題は残っているが，実用上の便利さはちょっとこたえられない．

Ⅲ. ひらきなおれば

プログラムの理解を容易にするため，あるいはプログラミングの見通しをよくするため，図を利用することはフローチャート等の例がある．ただし，これらはプログラムの制御の流れを表現するもので，本来，良い構造を持った言語に見易い字下がり（indentation）をつければ事足りるものでもあった．むしろ問題はデータ構造とその処理をどう図で表現するかである．実際，データ構造を図で示してプログラムの理解を助けることは極めて一般的な習慣であるが，ほとんどの場合それは単なる例示にとどまり，プログラム言語との融和性は高くない．その原因は，図は状態を表しはするものの操作をうまく表し得ないというところにある．

Ⅱでとりあえず述べた画法は，アセンブラで書いたプログラムを後から追いかけるために捏造したものであるが，そこに盛られている三つのアイデア

① 一枚の図の中に時間経過性を表現し，それによって操作を表わす．
② 一枚の図に並列的な処理を書き込んでしまう．
③ 図式表現とプログラム言語を融和させる．

は，もっと一般的なドキュメンテーション言語においても有用なものと考えられる．これが，最近とみに盛んになってきたデータ構造とその処理の表現法という問題の議論に，（いささか旧聞であるが）一石を投じることになれば幸いである．

Ⅳ. ありがとうございます

日頃御指導いただく池野信一特別研究室長に感謝いたします．

Ⅴ. 参考文献

日本治豊　教務事務の検証を目的とする小型電算機による帳簿組織（情処学会16大）