

実践報告・資料

コンパイラ制作のための教材開発

岐阜職業訓練短期大学校 宮田 利通

Supply of the Practical Teaching Book for Compiler Construction

Toshimichi Miyata

要約 情報工学での実践教育をめざした「物づくり」教材を開発した。開発した教材は、情報技術教育のための実技教科書「BASIC コンパイラの制作」である。この教科書は、コンパイラ理論とソフトウェア工学に基づいて模型のコンパイラを制作するための印刷教材である。

これは、主に職業訓練短大の実技指導書として開発した。しかしながら、向上訓練、短期実践研修などにも利用できる。コンパイラの設計は部品化を徹底したので、指導教官は、各種の教育訓練、受講者の能力、訓練時間の制約、コンピュータ・システムの違いなどに対し広範囲かつ柔軟に対応できる。

教材のコンパイラは、トップ・ダウンに制作が進行できるように設計されている。さらに、設計は構造化されているので、教材の利用者は、小さな単位の関数または手続きをコーディングして順次組み合わせていけばよく、容易にコンパイラを制作できる。教材は、コンパイラの機能を付加していくようにして制作が進むので、制作途中でもコンパイラの機能を確かめることができる。

コンパイラは、2パス・コンパイラである。パーサーは、LL(1)文法に基づく翻訳文法を使い、再帰下降解析を実行する。

1 はじめに

情報技術教育は、ハードウェアやソフトウェアの操作を中心とした教育から、原理を学び物づくりをめざした教育へと変化していくことが必要である。コンピュータが大衆に浸透するにしたがい、書店には啓蒙書やハウトゥ物があふれ、人々は流行のコンピュータやソフトウェアの評価談義に一喜一憂する。そして、これこそが情報技術であるとイメージしている。ファッションと技術教育は明確に区別しなければならない。

情報技術における物づくり教材は人変少ない。機械・電気・電子技術の分野は伝統があり、多くの人々により物づくり教材が提供されている。情報技術もこれらの伝統ある技術分野と肩を並べていけるように、地に足の着いた技術教育を進めていかなければならない。

実践教育をめざした教材として実技教科書「BASIC

コンパイラの制作」を開発した。この教科書は、ただ単にコンパイラをつくるのでなく、コンパイラ理論やソフトウェア工学で学んだ理論に基づき、実際に設計制作するものである。この教材を使うことにより、情報工学における理論が、物づくりにとっていかに重要であるかを学生は体得できる。

教材の開発環境は、汎用コンピュータ M-730/20 を使い、C 言語でコンパイラを記述した。

II 教材の概要

教材は、理論、実技シート、コンパイラのソース・プログラム等から構成されている。実技教科書の目次の部分を図1に示す。実技シートは、課題1から課題7より構成されている。学生は、この課題を順番に実行することでコンパイラを完成する。

目次

1. はじめに	1
2. コンパイラ	1
3. コンパイラのフェーズ	1
4. 文句解析	4
5. 構文解析	5
6. 記号表	6
7. コード生成	6
8. BASIC 文法と翻訳文法	7
9. 実技シート	16
部品系統図	17
部品表	18
大域変数の一覧表	23
課題1 フレームの構築	24
課題2 分岐ラベルの処理	28
課題3 枠組みの構築	46
課題4 代入文の処理	59
課題5 入出力文の処理	84
課題6 制御文の処理	100
課題7 サブルーチン文の処理	122
10. BASIC コンパイラのプログラム例	129
11. 目的プログラム例	167
12. 研究問題	172
13. 参考文献	173

図1 実技教科書の目次

教材は、トップ・ダウン開発をおこなうことで部品化を徹底した。これらの部品は、それぞれ独立したものであるから他の部品を構成する場合に何度でも利用できる。さらに、課題1から順番にコンパイラを作り上げていくことで各部品はその度にデバッグの対象となり、完成したときにはバグは少なくなっている。また、部品の機能が明確に作られているので、この教材を基にして学生は機能の拡張が容易にできる。

一つの部品につき一つの部品設計図を対応させた。部品設計図の実現には、IPO ダイアグラムを用いた。IPO ダイアグラムは、部品の入出力と機能を表現するのに使われる。しかし、これを基にプログラムをコーディングをするには論理構造が分からなければならない。一般に、論理構造を表現する手段として流れ図やPAD図が用いられる。これらの図式表現を使うと、図面が2枚に別れてしまい可読性と関連性が損なわれる。そこで、一枚の部品設計図で入出力、機能、論理構造を同時に表現するためにIPOダイアグラムの中で行動ダイアグラムを併記した。

行動ダイアグラムは、日本語で書かれたプログラム記述言語に論理構造を示す記号を書き込んだものである。枠型の図形を組み合わせた図式表現と比較して次

の特徴がある。

- (1) 狭い枠に文字を書き込まなくてすむので、十分に内容を表現できる。
- (2) コーディングされたプログラムと非常に近い表現形式で記述できるので、実際に即した実技指導ができる。
- (3) 行動ダイアグラムの記号は、デバッグの際にプログラム・リストに直接書き込めるので、設計図と対応が取れた実技指導ができる。
- (4) 記述するための用紙が小さくてすむ。
- (5) 特殊なテンプレートが不要である。

原始コードを目的コードへ変換するために必要な文法として翻訳文法を導入した。これは、LL(1)文法に翻訳をするときに必要となる情報や処理を付加したものである。教材の中でBASIC言語の翻訳文法を示した。学生がコンパイラを制作する場合は、この翻訳文法にしたがってプログラムを記述していけばよい。図2にBASICの翻訳文法を示す。

教材の翻訳文法を使うと構文解析を再帰下降解析でおこなえる。これは、文法とパーサーの対応が一致しているので学生にとり理解しやすい。すなわち、まず、

	生成用関	選択集合
<program>	→ code(PROGRAM) <line>* code(LABEL, X) <program body> END code(END)	{<line>}
<program body>	→ REX <comment> → <more lines> STOP code(STOP) → <more lines> LET <assignment> → <more lines> <assignment> → <more lines> INPUT <input list>* code(INPUT, X) → <more lines> PRINT <print list>* code(PRINT, X) → <more lines> GOTO <line>* code(GOTO, X) → <more lines> If <expression> <relative operator> <expression> code(POP-operator, use-operand, pop-operand) GOTO <line>* code(IF, pop-operand, X) → <more lines> FOR <assignment> DO <expression> <step part> code(FOR) → <more lines> NEXT <id>* code(NEXT, X, pop-label) → <more lines>	{REX} {STOP} {LET} {<id>} {INPUT} {PRINT} {GOTO} {IF} {FOR} {NEXT}

される2次元の表である。各要素は、外部名、内部名、変数の有無である。外部名は、原始コードに現れる識別子を格納する。内部名は、目的コードで用いられるラベル名である。これは、変数名に対してはVn、定数に対してはCn、一時領域に対しては@nの文字列で格納される。ただし、nは一連番号である。変数の有無は、論理値である。

ラベル表と記号表は、文字列を要素として持つ。さらに、演算子スタック、演算数スタックも文字列を要素に持つ。これは、学生が表の内容を知りたいとき、あるいは、コンパイルの進行状況を調べたいときに簡単に画面やプリンターにこれらの内容を出力できるからである。教材のコンパイラとして有効な活用が可能になる。

このコンパイラが出力する目的コードは、汎用コンピュータ M-730/20のアセンブリ語である。目的コードを出力する部品では、アセンブリ語の注釈を利用して、何の処理コードであるか、あるいは、何の変数や定数を参照しているか分かるようにした。これにより学生は、目的コードを見て原始コードがいかに翻訳されたかを容易に知ることができる。

IV 教材の内容

教材を使用しての実技は、実技シートにより実施できる。実技シートは、部品系統図、部品表、大域変数の一覧表、課題から構成される。課題は、7つの内容に分けた。学生は、課題1から順にこなしていくことにより、トップ・ダウンにコンパイラを完成できる。

部品系統図、部品表、大域変数の一覧表は、全体を表すものと、各課題ごとの部分を表すものと2種類からなる。これにより課題は、全体の中でどのような位置を占めるか理解しながら制作を進めることができる。図3に全体の部品系統図を示す。

課題1は、コンパイラのパスの制御をおこなう。すなわち、パス1がエラーを検出したらパス2は実行しない。さらに、コンパイラが異常終了したことを利用者に知らせる。

課題2は、分岐ラベルの処理をおこなう。この課題

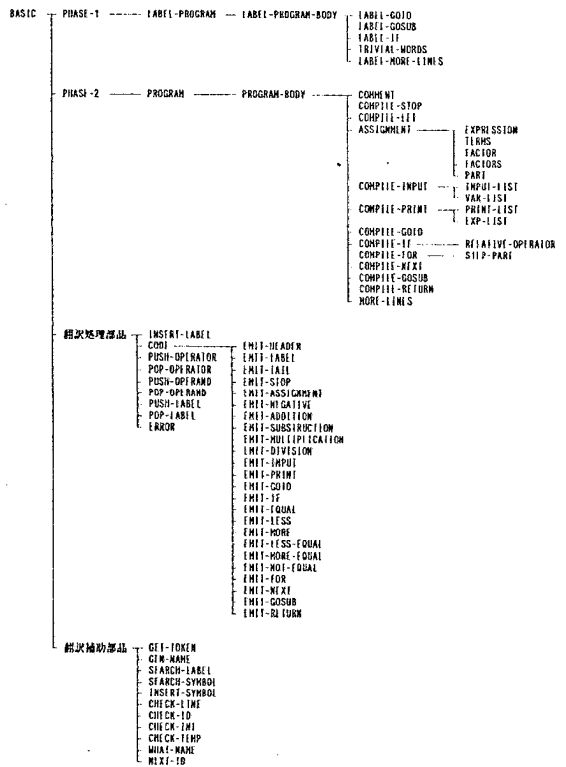


図3 コンパイラの部品系統図

では、字句解析ルーチンを作る。ラベルの処理には簡単な構文解析が必要なので、学生は再帰下降解析の概要が理解できる。

課題3は、パス2の枠組みを作る。目的コードのアセンブリ語における前処理と復帰処理を出力する。また、BASICのREM文とSTOP文の処理をする。

課題4は、代入文の処理をする。この課題が一番難しく、プログラムが最も長くなる。しかし、これを完成させれば後の課題は簡単に制作できる。

課題5は、入出力文の処理をする。この部分は、目的機械のオペレーティング・システムに依存するので、データ管理の知識が必要である。

課題6は、制御文の処理である。GOTO文、IF文、FOR文、NEXT文の処理をおこなう。IF文の条件部に現れる比較演算子の処理ルーチンを制作しなければならない。FOR文では、入れ子の処理のためラベル・スタックを用意する。

課題7は、サブルーチン文の処理をする。BASICは、引数の受け渡しや再帰呼び出しが無いのでGOTO

技教科書「BASIC コンパイラの制作」は、平成3年度職業訓練教材コンクールで入選した。教科書は、職業訓練研修研究センターで閲覧できる。

参考文献

- (1) A.V. Aho, R. Sethi, J.D. Ullman: Compilers Principles, Techniques, and Tools, Addison-Wesley Publishing Company, 1990
- (2) B. Pyster: Compiler Design and Construction, Van Norstrand Reinhold Company, 1980
- (3) J. McClure: Action Diagrams Clearly Structured Program Design, Prentice-Hall, Inc., 1985
- (4) P.M. Lewis, D.J. Rosenkrantz, R.E. Stearns: Compiler Design Theory, Addison-Wesley Publish Company, 1976
- (5) 国友義久: 効果的プログラム開発技法、近代科学社、1988