OS/omicron における
2B-7 プログラミング言語のCAIの基本設計

森田雅夫、並木美太郎、高橋延匡
(東京農工大学)

1．はじめに

プログラミングの教育は、ただ単に知識を詰め込めばよいわけではない。文法を覚えただけでは、プログラムを書けるようになったとは言わない。プログラミングをする場合には、なんらかの問題をプログラムで解決しようとするわけであるが、その際に、
(1)固定概念にとらわれず、問題をさまざまな面から観察できる。
(2)問題点を的確にとらえることができる。
(3)解決策をプログラム言語で記述できる。
これら3つの要素がプログラマに必要である。

実際のプログラミングでは、(3)は特定の言語で記述する必要はなく、仮想的なプログラミング言語でもよいのである。解決策(＝アルゴリズム)をプログラミング言語と自然言語を混在した形で記述するのも一つの方法である。したがって、プログラマに必要な要素のうち、重要なのは(1)(2)のプログラミングのセンスである。

プログラミングのセンスは、システム開発の経験により育成されるが、そのような経験者が常に教育を行えるとは限らない。そこで、プログラミング言語のＣＡＩの必要性が出てくる。特にコンピュータ教育の場合、実務もコンピュータ相手であるので、簡単な仕事の例を問題にして教育していけば、教育と開発環境になれることができる。

プログラミング言語に限らずどんな言語を習得する場合にも、“習うよりは慣れろ”とよく言われる。この点から見ても、プログラミング言語のセンスを磨くには、机上での講義より、演習形式の方が効果があるのは言うまでもない。したがって、ＣＡＩでも演習の中で生徒を補助する形式がよいと思われる。特にプログラムを実行させることに重点を置き、実行時のエラーに強いシステムであることが条件である。とにかく様々なことをユーザに実行させ、その結果に対する解説、評価を行なうことで、ユーザのプログラミングのセンスを磨いてことができる。

プログラミング言語のＣＡＩは、プログラミング教育の一形態として、ユーザが気軽にコンピュータを使える環境を提供する、という観点から必要であると考えられる。

2．プログラミング言語のＣＡＩに要求される機能

実際に人間がプログラムを作成するときの行動様式を考察した。プログラミングをするときの行動・思考の流れを図1に示す。

ここで、“文法エラー”とは、コンパイラ言語であれば、パーザ、リンカなどでエラーが発生し、実行プログラムが生成されなかったときの事を指す。また、“アルゴリズムエラー”とは、実行プログラムが生成されても、実行結果が正しくないときの事を指す。

人間がプログラミングするときには、５つのステップがあることがわかる。
(1)解決すべき問題が起こる。
(2)プログラミング言語を決める。
(3)データ構造とアルゴリズムを決める。
(4)アルゴリズムをプログラミング言語で記述する。
(5)プログラムの実行・評価をする。

(1)(2)(3)はプログラミング言語には依存せず、解決すべき問題の知識と情報工学の知識を必要とする。しかし、(4)(5)は選択したプログラミング言語に依存した、

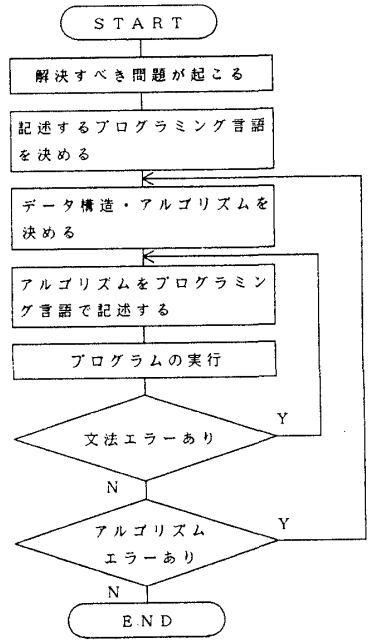

図1　プログラミングの流れ

言語固有の知識を必要とする。このように考えると、この５つのステップは大きく２つの部分に分けられ、(1)(2)(3)は基本設計、(4)(5)はプログラミング言語による実現として捉えることができる。

演習形式で学習を進めるプログラミング言語のＣＡＩにも、この５つのステップはあてはまる。プログラミング言語のＣＡＩを用いた場合の学習の流れを図２に示す。

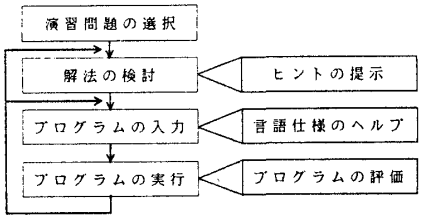

図２　ＣＡＩを用いた場合の学習の流れ

図２の中の演習問題の選択、解法の検討、プログラムの入力、プログラムの実行が、それぞれ(1)(3)(4)(5)に対応する。ＣＡＩの場合にはプログラミング言語は決まっているので、(2)は除かれる。

演習問題の選択はユーザに任されるが、それまでの学習結果を考慮し、システムが提示することも可能である。これは、コースウェアの設計の話なので今回は省略する。

Basic Design of a Programming Language CAI on OS/omicron
Masao MORITA, Mitarou NAMIKI, Nobumasa TAKAHASHI
Tokyo University of Agriculture and Technology

・解法の検討に対する支援
　問題のヒントを表示する。ヒントも何種類か用意し、作成されたプログラムのレベルや、それまでの学習の結果により変える。ヒントは、学習者から要求があったときのみ表示するヘルプ形式がよい。
・プログラムの入力に対する支援
　言語仕様をオンラインマニュアルの形で表示できるようにする。これもヘルプの形式にする。
・プログラムの実行に対する支援
　エラーの解説をする。実行の結果により、プログラムの入力または、解法の検討に戻る。
　エラーには文法エラー・実行時エラー・プログラム構造のエラーがあると考える。各種エラーに対して、ＣＡＩシステムは次のような機能強化で対応する必要がある。
(１)文法エラー
　ソースプログラム中のエラーの発生した位置を明確にし、なぜそのエラーが発生したのかを解説する。システムから見れば、その位置には何がなければならないのかを表示できるのが望ましい。
(２)実行時エラー
　ソースプログラム中のエラーの発生した文を明確にし、エラーの原因を解説する。ソースプログラムを基準に解説することが重要である。
(３)プログラム構造のエラー
　プログラミングの初心者は、無駄のあるプログラムを書きがちである。これはプログラミングのセンスが悪いからである。そこで、プログラム構造の悪いところをチェックし、どのように直せばよいかを指摘する。

３．初心者の陥りやすい間違いと初版システムの実現

　このシステムの対象とするユーザは、当研究室の４年生とし、対象とする言語は言語Ｃに設定した。理由としては、自分が当研究室に所属しているので、実体験に基づいた指導が可能になるからである。
　システムを作成する場合には、ユーザの把握をしなければならない。実体験、すなわち教師と生徒を重ね合わせることにより、次の点について自分の経験を生かせるようになる。
　(１)ユーザはどの程度プログラミングに関する知識を持っているのか。
　(２)ユーザはどのようなことが理解しにくいのか。
　(３)ユーザは何を知りたがっているのか。
　そこで、今年度は研究室の学生、教官を対象に言語Ｃを用いたプログラミングでどんなミスをしたのか調査を行なった。その結果によれば、ポインタによるミスが一番多い。ポインタによるミスは、ポインタ変数が何を指すのか分からない、という理由によるミスである。したがって、プログラム中のポインタ変数の構造を図示すれば、教育効果が大きいのではないかと考えた。そこで、プログラムのデータ構造を図示する機能をまずはじめに実現することにした。
　また言語Ｃは、記述に関してユーザに任されている部分が多い。したがって、同じアルゴリズムをプログラムで記述しても個人差が大きくなる。そこで、最適化の手法を用いて、ユーザプログラムの冗長な部分をチェックしようと考えている。

４．チェックの実例

　データ構造を図示した例を図３に示す。図３は、構造体listのメンバdataの内容を表示した例である。ポインタ変数はメモリを表す箱と、参照を表す矢印で示される。まだ実体がないポインタ変数は、箱と矢印しか表示されない。(図３のＩの部分)実体のあるポインタ変数は、矢印の先に別の箱が表示される。(図３のIIの部分)箱がある限りは、矢印を伝って、その中身を調べていくことができる。
　このようなデータ表現は、言語Ｃのプログラマならば誰もが知っており、データ構造を考える場合には、誰もが一度は書く図である。初心者はデータ構造のメモリ・イメージを把握しにくいので、その手助けをする意味も含んでいる。
　プログラムの冗長な部分を指摘した例を図４に示す。図４は、不用な一時変数と代入がある場合を示している。解説はソースプログラムとは別のウィンドウに表示され、簡単に移動できる。

```
struct sample {
    struct sample *before;
    struct sample *next;
    char *data;
    int len;
} *list;

static char buf[5] = "name";
list->data = (char)malloc(sizeof(buf));
strncpy(list->data, buf, sizeof(list->data));
```

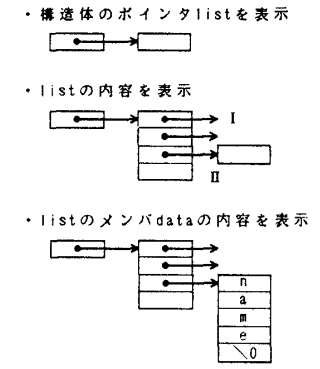

図３　データ構造の図示の例

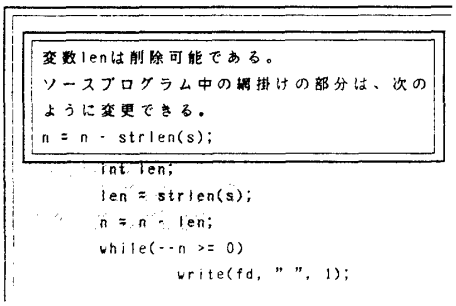

図４　プログラムのチェック例

　初心者は一時変数を多用する傾向があるが、言語Ｃでは式自体が値を持つので、一時変数を使わなくてもすむ場合があるのである。最適化の手法を用いて、不用な一時変数の消去をしたり、条件判定を変えることにより制御構造が簡単になる場合などをチェックする。

５．終わりに

　本論文では、プログラミング言語のＣＡＩに要求される機能を示した。現在、ＣＡＩシステムのうちプログラム・チェック機能を作成中である。

参考文献
１）鈴木茂夫、他：言語Ｃシステムプログラマ養成のためのインテリジェントＣＡＩの設計、情報処理学会第34回全国大会(1987)、４Ｈ－２
２）大槻、竹内：ＩＣＡＩに要求される機能と構造について、情報処理学会第31回全国大会(1985)、８Ｈ－１