【132】

氏名	渡辺勝正（わたなべかつまさ）
学位の種類	工学博士
学位記番号	工博第176号
学位授与の日付	昭和44年9月24月
学位授与の要件	学位規則第5条第1項該当
研究科・専攻	工学研究科数理工学専攻
学位論文題目	**Compiler Compiler に関する研究**
論文調査委員	（主査）教授 萩原宏　教授 清野武　教授 樋木義一

論文内容の要旨

この論文は計算機のソフトウエアの重要な部分をなすCompilerについて，言語理論の面から検討を加え，プログラミング言語 ALGOL の研究とその Compiler 作成の経験に基づいて，Compiler 記述言語 COL を構成し，これを用いて作成した Compiler Compiler について考察を加えたものであって，7章から成っている。

第1章は序論で，まず，言語の基本となる記号列の性質を調べ，書きかえ規則で規定される句構造文法について論じている。つづいて，Comipler の働きについて検討を加え，Compiler を特性づける三つの言語，すなわち，入力言語，出力言語，Compiler を記述している言語のそれぞれの観点から Compiler Compiler の方式を比較検討して，著者は Compiler 記述言語を用いて Compiler を記述する方式を採用したことを述べている。

第2章では記述される Compiler の入力言語のモデルとして ALGOL 60をえらび，ALGOL（一般にプログラミング言語）の様々の概念がいくつかの基本概念によって表現されることを考察している。これは入力言語の記号列がどのような出力列に変換されるかという問題の基盤を与えるものであり，Compiler の基本能力として何が必要かを示すものである。ついで，ALGOL における block 構造に基づいた名前（identifier）の対応づけ，Procedure 構造に基づいた記憶の割り付けの二つの問題に対する方策を論じている。

特に，後者については recursive prcceure に対処するため dynamic storage allocation の方法を考えている。

第3章では，入力言語の構造分析法として最も形式化しやすい Syntax-Directed Analysis について，その二つの方法を比較検討している。すなわち，入力記号列の構造を表わす Tree に対応して，Top down 分析法，Bottom up 分析法と呼ばれるものであって，後述の COL で記述される Compiler における構造分析法の基礎となるものである。

第4章では，著者が構成した Compiler 記述言語 COL について述べている。COL は Compiler の構造分析過程を記述する Syntax statement と意味解釈過程を記述する Semantic statement とから成っている。COL に導入されている変数および Statement は Compiler，とくに ALGOL Compiler に必要不可欠な機能を表わすものであり，direct execute machine に対して一つの示唆を与えるものと考えられる。Compiler の二つの過程の間には入力記号列の構造を表わす M Structure が導入される。すなわち，M Structure は入力記号列をある標準型に表現しなおしたものであって，意味解釈を容易にして，その手順を簡潔にするためのものであり，M Structure からもとの入力記号列が復元可能である。

構造分析過程から意味解釈過程への移行は入力記号列全体の構造分析が終ってからなされるのが普通であるが，COL では構造分析の任意の時点で意味解釈過程に移る指令を出すことができる。一般に，その指令は入力記号列のある分脈の構造が確定して分析をやり直す必要がなくなった時点で出される。

第5章では，まず，COL が記述される Compiler が受け入れることのできる入力言語の性質を検討し，Syntax statement で表現される構造分析手順が LR(k) 文法を受け入れることができることを示している。また，Semantic statement は言語の Semantics を形式的に与えるもので，Compiler の意味解釈過程を表現することができることを述べている。

COL では現在の計算機を対象にし，出力列は具体的な機械語あるいは簡単な記号言語で表現されるものとしている。すなわち，入力記号列が M Structure に変換されて，M Structure と出力列との対応づけが Semantic statement によって記述されるのである。

また，Landin の AE (appricative expression) と SECD machine による Compiler の抽象的な考え方を検討し，COL と比較している。

最後に COL で記述される Compiler は push down automaton と stack automaton を連結した2連 automaton にモデル化されることを示し，CF 言語を入力言語として受け入れることができることを示している。

第6章では COL で書かれた Compiler を機械語に変換し，計算機の中に Compilerを作り出す COL processor について述べ，それによって作成された三つの Compiler について検討を加えている。それらは ALGOL の subset およびほぼ full ALGOL に近いものを入力言語とし，記号言語あるいは機械語を出力言語とするものである。Compiler 記述上，構造分析の単位の選び方と出力記号の指定の仕方に特に注目して，それらの意味解釈過程の記述におよぼす影響を論じている。

これによって作成された Compiler と hand-coding による Compiler とについてコンパイル時間を比較してみると，簡単な Compiler で1.5倍ないし2倍，最も複雑な Compiller で4ないし6倍要している。

第7章は結論で Compiler 記述言語として著者が構成した COL についての問題点を検討し，一般に Compiler 記述言語による Compiler Compiler の今後の問題点について論じている。

論文審査の結果の要旨

計算機の発達に伴い，種々のプログラミング言語を用いた自動プログラミング方式の採用によって計算機の利用は容易となり，需要が増大し，更に種々の問題に応じて様々なプログラミング言語が開発されて

いる。しかし，それらを利用するためには，Compiler 作成という多大の時間と労力を要する作業が必要となる。新しい計算機，新しいプログラミング言語が開発されるにつれ，必要な Compiler はますます多くなる一方である。そこで Compiler 作成の道具として計算機を利用し，時間と労力を軽減しようとする種々の試みがなされている。

本研究は，Compiler の入力言語と出力言語を定義し，対象とする計算機の仕様を与えることによって Compiler を作り出す Compiler Compilerを作成することを目的に行なわれたものである。

まず，言語の基本となる記号列の性質について調べ，句構造文法について論じ，Compiler を特性づける三つの言語（入力言語，出力言語，Compiler を記述している言語）のそれぞれの観点から Compiler Compiler の方式を検討し，Compiler 記述言語を用いて Compiler を記述する方式を選んで研究を進めている。

入力言語のモデルとして ALGOL 60をえらび，プログラミングの言語の様々の概念がいくつかの基本概念によって表現されることを考察して，Compiler の基本能力として何が必要かを示している。

つぎに，入力言語の構造分析法とし，最も形式化し易い Syntax-Directed Analysis の二つの方法，すなわち，Top down 分析法および Bottom up 分析法について比較検討している。

著者は Compiler を記述するための言語として，COL (Compiler oriented language) を構成している。これは Compiler の構造分析過程を表現する Syntax statement と意味解釈過程を表現するための Semantic statement から成っている。これによって作成された Compiler は，入力言語の記号列を受け入れてその構造を分析して結果を表わす M Structure を構成する部分と，これに基づいて意味解釈を行ない，目的とする出力列を構成する部分とに分かれる。

ついで，COL で記述される Compiler が受け入れることのできる入力言語の性質を検討し，Syntax statement で表現される構造分析手順が LR (k) 文法を受け入れることができることを示し，また，Semantic statement が言語の Semantics を形式的に与えるものであることを論じ，これらを合せて Compiler 全体を2連 automaton にモデル化して COL が CF 言語を受け入れることができることを結論している。

更に，著者は COL で記述された Compiler を機械語に変換する COL processor について述べ，それによって作成された三つの Compiler について検討を加えている。それらは ALGOL の subset および full ALGOL に近いものを入力言語とし，記号言語あるいは機械語を出力とするものである。

最後に著者の COL による Compiler 作成の経験を基にして Compiler 記述言語による Compiler Compiler の問題点について考察を加えて，Compiler Compiler 作成に当って注意すべき点を明らかにしている。

以上を要するに，本論文は Compiler の必要とする基本能力を明らかにし，入力言語の分析法について検討を加え，新たに Compiler を記述するための言語として COL を構成し，これによって実際の計算機に対する Compiler を作成し，その結果をもとにして Compiler Compiler の問題点を明らかにしたものであって，学術上，実際上寄与するところが少なくない。

よって，本論文は工学博士の学位論文として価値あるものと認める。