



Title	LR(k)アナライザ及びマイクロプログラム記述言語に関する研究
Author(s)	菊野, 亨
Citation	大阪大学, 1975, 博士論文
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/1023">https://hdl.handle.net/11094/1023</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

氏名・(本籍)	菊 <sup>きく</sup> 野 <sup>の</sup> 亨 <sup>とあら</sup>
学位の種類	工 学 博 士
学位記番号	第 3 3 7 0 号
学位授与の日付	昭 和 50 年 3 月 25 日
学位授与の要件	基礎工学研究科物理系 学位規則第5条第1項該当
学位論文題目	LR(k) アナライザ及びマイクロプログラム記述言語 に関する研究
論文審査委員	(主査) 教 授 嵩 忠雄 (副査) 教 授 田中 幸吉 教 授 藤沢 俊男 教 授 木沢 誠

## 論 文 内 容 の 要 旨

本論文はコンパイラの語い解析と構文解析の方法、および、マイクロプログラム記述言語の設計とそのトランスレータの作成法に関する研究を2編にまとめたものである。

よく引用される例であるが、フォートランのDO250 I = ... は文脈を利用しなければ、トークン名(整数、変数名などといった基本的なカテゴリの名前)とその区切りの位置を決定できない。=のつぎがたとえば13, ...であればDOがdo文を示すキーワードであるが、13. ...ならDO250 Iが実数型の変数名である。本論文の第1編第1章では、このフォートランの例を含むように、つぎに来得る長さh+1のトークン名の系列をLR(k)パーザから教えてもらい、それを利用して文字列をトークンごとに区切るという方法を考えている。一般には、hを長くしてh+1個先のトークンまで一致しても、その一番目のトークンの区切りに関するあいまいさがおこりうる。そのため、h+1個先のトークンまで見るならば一意に決まるという条件をつけ、この条件がなりたつときh-区分可能であるという。

本論文では、hを指定すればh-区分可能かどうかは判定できるが、hを与えなければそれは判定不能であることを証明している。さらに、区切りの可能性を一時記憶するレジスタの数が最小のスキャナの構成法を示し、本論文で導入するLR(k)パーザとスキャナのモデルで受理する言語が決定性言語になることを証明している。

パーザは、通常、特定の文法Gに対し、与えられた入力系列がそのGで導出されるかどうか判定し、もし導出されるならその導出木を出力する。ところが、コード発生などの意味づけの処理をするとき、固定したもとの文法Gにおける導出木そのものは必ずしも必要でない。本論文の第1編第2章では、構文解析部の本質的な機能を、わたすべきパラメータを定めてセマンチックルーチンを正しく呼ぶこ

とであるとし、この機能を果すものをアナライザと呼ぶ。アナライザのモデルとして、一定量までの先読みを許した決定性プッシュダウンオートマトンを考える。二つのアナライザは同じ入力系列に対して、同じパラメータで同じセマンチックルーチンをつぎつぎと呼ぶとき、等価と定義する。

本論文では、任意のアナライザが与えられたとき、それと等価なアナライザのうちで、先読みの長さが常に最小のアナライザを求める方法、さらに、先読みが最小のものの中でもっとも簡単な（状態数が最小の）アナライザを求める方法などを示している。

本学情報工学科にある小型電子計算機 UX は16ビット長の垂直型のマイクロ命令を備え、書き換え可能な4k 語の制御記憶をもったマイクロプログラム方式の計算機である。マイクロプログラム記述言語としてマイクロアセンブリ言語がメーカから提供されているが、ハードウェアにかなり依存しており、マイクロプログラムを書くには、ハードウェアの細部に関するかなりの知識を必要とする。そのため、マイクロプログラム記述用的高级言語の開発が望まれている。

本論文の第2編では、この UX を対象として設計したフローチャート型のマイクロプログラム記述言語 FML の仕様と、そのトランスレータの作成法およびそこでのいくつかの問題について述べている。この言語のおもな長は「ユーザが使いやすく、コーディングが見やすい」ことである。マイクロプログラムの効率は特に重要であるので、トランスレータでは種々の最適化を試みている。

## 論文の審査結果の要旨

語い解析と構文解析はコンパイラの主要な機能である。本論文では、まず従来の語い解析の方法を拡張して文脈を有効適切に利用する方法を提案し、この方法が適用可能な文法のクラスが実用上十分広いことを示し、またこの方法を実現する場合に必要な記憶容量を最小にするアルゴリズムを示している。ついでコンパイラの構文解析部の適切なモデルとして、アナライザを導入し、その等価性の判定法、等価なアナライザのなかで先読みが最小であるものを求める方法、さらに状態数最小のものを求める方法を示している。これらは従来のパーザの単純化に比べて、より一般的で効果的である。本論文の後半において、フローチャート形のマイクロプログラム記述言語の仕様と、そのトランスレータ作成法およびマイクロプログラム特有の問題点と解決法を示している。これらの結果は、コンパイラの理論と作製法に関して、新しい重要な知見を加えたものであり、高く評価される。