

プログラム仕様に用いる自然語の処理システム

— 代数的仕様への変換と知識管理 —

石木 信之*

並河 英二*

関 浩之*

杉山 裕二*

藤井 護**

鳥居 宏次*

中小路 久美代***

*大阪大学基礎工学部

**大阪大学大型計算機センター

***株式会社ソフトウェア・リサーチ・アソシエーツ

自然語の分野をプログラム仕様に用いられるものに限定し、その意味を代数的枠組の中で定義しようとする試みがなされている。そこでは、自然語文を適当な論理体系における論理式に変換し、各語句の意味を公理で定義し、自然語文の意味は対応する論理式のその論理体系における意味であるとしている。本報告では、この方式に基づいた自然語処理システムを紹介する。システムとしては、自然語文を論理式に変換するプログラム、および語句とそれを意味定義する公理との対応を表わす辞書が必要である。自然語の分野を限定したとはいえ、その語句の数は膨大なものになると予想されるため、自然語文を論理式に変換したり、公理を辞書に蓄積するシステムとしてだけでなく、それらを効率的に検索、更新するための支援システムとして、本システムを作成した。

A Processing System of a Natural Language Which Is Used for Writing Program Specifications

Nobuyuki ISHIKI*, Eiji NABIKI*, Hiroyuki SEKI*, Yuji SUGIYAMA*,
Mamoru FUJII**, Koji TORII*, and Kumiyo NAKAKOJI***

*Dept. of Information and Computer Sciences, Faculty of Engineering Science, Osaka University
(1-1 Machikaneyama, Toyonaka, Osaka 560, Japan)

**Computation Center, Osaka University

***Software Research Associates, Inc.

A method to define English which is used for writing program specifications is proposed. In the method, a sentence in English is transformed into a formula in a formal logic, and a meaning of each phrase is defined by axioms. This paper describes a natural language processing system based on the method. Even if a field of a natural language is limited to writing program specifications, the number of phrases is expected to be vast. So, this system is constructed as not only a system to transform a sentence into a formula, but also a system to support increasing a dictionary.

1. はじめに

自然語の分野をプログラムの仕様記述に用いるものに限定して、その意味を代数的枠組の中で定義しようとする試みがなされている⁽⁴⁾⁽⁵⁾。そこでは、自然語の仕様記述能力は保ちつつできるだけ限定した部分言語(以下、この部分言語を単に自然語と呼ぶ)を決め、その構文則を定義し、さらに、自然語文から(適当な論理体系における)論理式へ変換する変換則を定めている。語句の意味を公理で定義し、自然語文の意味は、変換された論理式のその論理体系における意味であるとしている。各論理式自体の意味は代数的に定義することができる。

この方針の特徴としては、自然語文の意味を厳密に定義することができる、書かれた仕様を形式的に議論することが容易である、といった点が挙げられる。また代数的言語ASLについては検証支援系が作成されており⁽²⁾、さらにASLの部分言語である関数型言語ASL/Fや抽象的順序機械ASMについては実行系も作成されており⁽¹⁾⁽³⁾、抽象的な仕様に基づいて作られた(ASL/Fなどによる)実行可能プログラムが元の仕様を満たしているかどうかの形式的議論も厳密にかつ計算機の支援の下に行なうことができる。そこで自然語仕様を代数的仕様へと変換するシステムを作成することになった。

システムとしては、自然語文を論理式へと変換するプログラム、および語句と公理の対応を表わす辞書が必要である。しかし、自然語の分野を限定したとはいえ、その語句の数は膨大なものになると予想されるため、辞書の作成は各種の例題を通して徐々に行なっていくという方針を採った。そのためには、単に公理を辞書に蓄積するだけでなく、それらを効率的に検索、更新するための支援システムも必要である。

本システムは、自然語で書かれた仕様を論理式の仕様へ変換するシステムであると同時に、辞書作成の支援や蓄積された知識の管理も行なうシステムである。

2. システムの概要

本システムの行なう処理は大きく分けて、自然語文の構文解析、構文解析された自然語文の論理式への変換、辞書の管理の3つの処理に分けられる。ユーザは、本システムのコマンド・インタプリタを通してこれらの処理を行なう。

また、自然語の意味定義においては、辞書としてこれだけの情報があれば十分であるといった目安が確立されていないため、辞書に対して、得られた情報の損失を極力小さくしたり、新たな情報を追加できることが望まれ

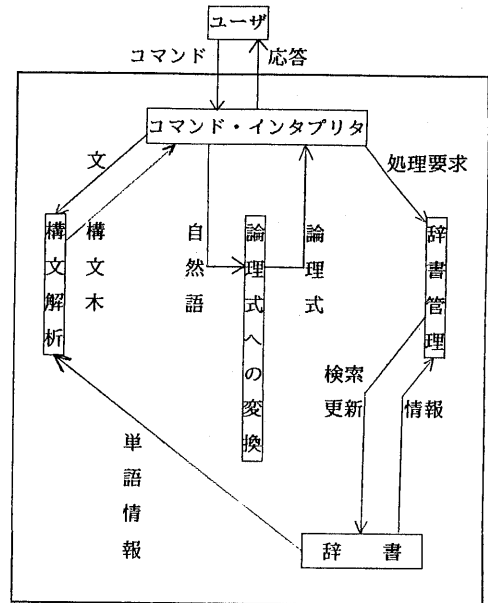


図1 システム構成図

る。それについて本システムでは、辞書作成の際、得られた情報はそのまま記憶する他、別の情報として関連語句も記憶できるようになっている。

また、本システムは、オプションを含めたコマンド名の変更やコマンド・ファイルの使用を可能にするなど、ユーザ・インタフェースに関しても考慮がなされている。

2.1 構文解析

自然語文の構文解析を行なう部分であり、この部分は自然語の構文則をPrologのDCGで記述したものである。

自然語では、文によっては構文木が一意に決定できない場合もありうる。そのような場合、本システムは考えられる構文木を全て表示し、その中からユーザが意図したものを選択するといった方法をとる。

2.2 論理式への変換

与えられた自然語文を、定められた変換則により論理式へと変換する部分である。

これは、自然語文の構文木から対応する論理式の構文木を生成するものであるが、1つの自然語文から複数の論理式が考えられる場合もあり、そのような場合は、構文解析部同様、考えられる論理式の構文木を全て表示し、ユーザが選択するといった方法をとる。

なお、具体的な構文則や変換方法については、文献(5)を参照して頂きたい。

2.3 辞書の管理

本システムの持つ辞書を管理する部分である。ユーザはこの部分を用いて、辞書の更新や検索を行なうことができる。

辞書には、構文解析のために単語に関して品詞、格、時制など文法上の情報を格納しておく部分と、語句の意味定義となる公理を蓄積する部分とがある。本システムでは、構文解析部をprologで記述しているため、構文解析に使用する単語の情報をprologの単位節で記述する。このため、単語に関する部分と公理に関する部分とを1箇所にまとめることができず、単語の辞書と公理の辞書とを別々に考えることにする。また、辞書管理部も単語辞書管理部と公理辞書管理部に分けて考える。

単語辞書は各単語に関する文法上の情報を格納しておくものであるが、本システムの対象となる自然語の部分言語も一度に決定されるべきものではなく、逐次語彙が拡張されていく必要がある。単語辞書管理部は、単語辞書に単語を追加したり、検索するための手段を提供するものである。これにより、ユーザは、自然語の構文則やDCGに関する予備知識なしに、単語辞書を扱うことが可能である。

公理辞書は、自然語で書かれた公理を、対応する論理式とともに蓄積しておくものである。しかし、ユーザが知りたい語句やそれを定義した公理を効率的に検索できるようにするためには、語句や公理を単に表の形で羅列して記憶だけでは不十分である。

そこで、本システムの公理辞書は、公理の他に、それによって定義される語句や、その語句の関連語句、見出し語といったものも記憶できるようになっている。

ユーザは、この公理辞書管理部により、公理辞書を更新したり、語句を検索する際には、その語句を構成する単語から検索したり、別の語句との関連から検索することができる。また、更新、検索する語句を指定する時には、特殊記号を用いて、同時に複数の語句を指定することも可能である。

なお、本報告では以下、公理辞書、公理辞書管理を、それぞれ単に辞書、辞書管理と呼ぶ。

2.4 コマンド・インタプリタ

ユーザは、このコマンド・インタプリタを通して、各処理を行なうことができる。また、このコマンド・イン

タプリタは、ユーザ・インタフェースを考慮し、UNIX上のshellの機能を一部参考にして作られており、コマンド・ファイルの実行や、alias機能を用意している。

3. 公理辞書の構成

辞書は、意味定義された語句やそれを定義する公理を格納するものであるが、一般に語句は複数の公理により意味定義されるものであり、また、1つの公理文が複数の語句の意味定義に用いられることもある。さらに、語句とその意味を定義する公理との対応を記憶しただけでは、例えば、ある語句を検索する際、ユーザが語句自体を知らない時にその語句を構成している単語から検索する、といった方法が困難である。

そこで、本システムでは、語句や公理を単に表の形で記憶しておくだけではなく、意味定義されている語句そのものの以外からも検索できるように見出し語を付けたり、ある語句からその関連語句も知ることができるように語句間の関連状況も記憶することが可能になっている。

そのため、本システムの辞書は、項目、語句関連グラフ、公理、の3つの部分から構成されている。

3.1 項目

項目は、辞書の見出しに相当し、アルファベット順に記憶される。また、各項目は、それから検索できる語句をリストとして記憶している。

項目名は、単なる見出しであり、必ずしも定義される語句そのものである必要はない。例えば、連語の場合、その連語自体だけでなく、連語を構成する単語も項目名となりうる。

これにより、ユーザが辞書を検索する際、定義された語句そのものを知らずとも、語句を構成する単語から検索することが可能である。また、項目名がアルファベット順に格納されているので、いかなる語句が定義されているかを知ること容易である。

3.2 語句関連グラフ

意味定義された語句の間には、何らかの関連が付けられるものがある。例えば、集合を表す語句とその元を表す語句、または、集合を表す語句とその部分集合を表す語句、などの関連をもつものがある。そういった語句の間の関連も辞書に保存できれば、語句を効率よく検索することができるであろう。本システムは、実際に意味定義されている語句、および、語句間の関連状況を有向グラフを用いて記憶する。

意味定義された語句には1つつグラフの頂点を割りつけ、その語句をその頂点のラベルとする。語句間の関連はグラフの有向枝で表す。さらに、その枝には、それがどのような関連であるかを表す関係名が付いている。また、すべての関係にはその逆関係も存在する。従って、語句Aから語句Bへの関係が付けられた時には、同時に語句Bから語句Aへ、その逆関係が付けられる。

項目が単なる見出しであるのに対し、グラフの頂点にラベル付される語句は、実際に意味定義されている語句である。語句は構文木の形でラベル付けされる。また、各語句は、それを定義する公理をリストとして持っている。

これらにより、ある語句がどのような公理を用いて定義されているか、また、ある語句の関連語句としてどのような語句が定義されているか、例えば、ある集合を表す語句に対して、その部分集合としてどのようなものが定義されているか、などを知ることができる。

語句関連グラフの例を図2に示す。

3.3 公理

グラフの頂点のラベル名となっている語句の意味は、公理を用いて定義される。公理には、自然語で書かれたものとそれを変換した論理式とがある。

また、公理は、その内容によって種類分けすることができる。本システムでは、各公理に対して、それがどのような種類の公理であるかを表す属性を付けることができる。例えば、同義語において、ある語句をすでに定義さ

れている別の語句で言い換えることによって定義するような場合に、その公理にそれが言い換えのための公理であることを示す属性を付けて、他の公理との働きの違いを明示することができる。

本システムでは各公理について、自然語文、論理式、属性の他に、最初にその公理を記述したユーザの名前および日付、時刻、さらにその公理が記述された後訂正された時には、最後に訂正したユーザの名前、日付、時刻も記憶する。これらにより、ある働きをもつ公理を取りだしたり、あるユーザによって定義された公理を取りだしたりすることが可能である。

4. 公理辞書管理プログラムの機能

前章で述べたように、本システムの辞書は3つの部分から成っている。ユーザは、それらの部分をそれぞれ独立に追加、削除、変更することができる。また、検索の際には、コマンドにオプションを付けることで、不必要な情報の表示を防ぐことができ、効率的に検索を行なえるようになっている。削除の際にも、必要な情報を誤って削除することのないよう、コマンドにオプションを付けることにより削除に先立ってユーザに確認を求める機能も持っている。さらに、一部のコマンドを除き、ワイルドカード文字などの特殊記号が使用できるので、ユーザは検索する語句などを指定するときに、それを完全な形で指定する必要はない。

ここでは、本システムの持つ辞書管理機能を、コマンドを交えて説明する。

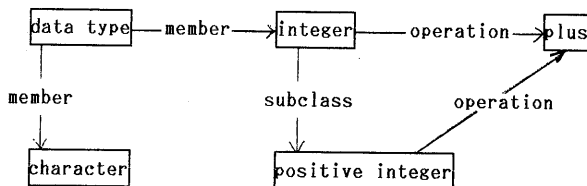


図2 語句関連グラフ

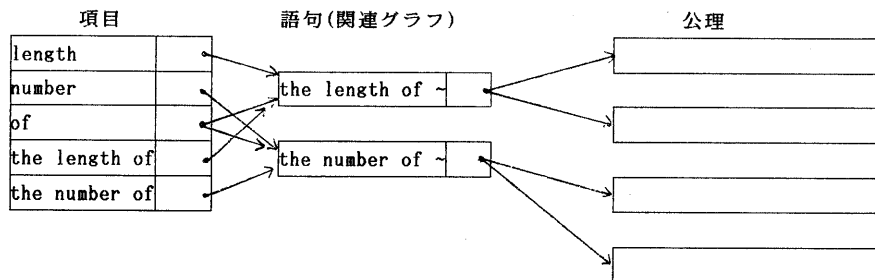


図3 項目、語句、公理の関係

4.1 項目に関する機能

対象となる項目を指定するには、項目名による場合と、項目指定子による場合とがある。

項目名は、辞書に格納する、またはすでに格納されている項目の名前であり、それ自身で項目指定子となる。

項目指定子は、項目名の一部または全部を後述の特殊記号で置き換えた形をしており、1個の項目指定子で複数の項目を指定することができる。

a) 項目の登録

mki 項目名₁ [項目名₂ … 項目名_n]

登録する項目名をコマンドの引数に並べることで、一度に複数の項目名を登録することができるが、同じ項目名を複数登録することはできない。

各項目名は、それが登録された順序に関係なく、アルファベット順に格納される。

b) 項目の削除

rmi [オプション] 項目指定子

オプション '-i' を付けることで、削除される項目を選択し、確認することができる。

c) 項目名の変更

chi 旧項目名 新項目名

変更された項目は、新しい項目名によって、整列された位置に格納される。

d) 項目から検索できる語句の宣言

refer 項目指定子-語句指定子

これにより、指定された項目から指定された語句を検索することが可能になる。

1つの項目から検索可能な語句の間には相対的な優先順位が存在し、新たにこの宣言がなされた語句は優先順位の最も高いものとなる。この優先順位は、4.4 で述べる機能を使用する際に意味を持つ。

語句指定子については、4.2 語句に関する機能 を参照して頂きたい。

e) 項目の表示

書式1: pri [オプション] [文字列₁] > [文字列₂]

書式2: pri [オプション] 項目指定子

オプション

-c 表示前に、ウィンドウを消去する。

-p 各項目について、その項目から検索できる語句も表示する。

書式1を使用すれば、文字列₁で始まる項目名を持つ項目から、文字列₂で始まる項目名を持つ項目までを表示させることができ、書式2を使用すれば、項目指定子に

よって指定される項目を表示させることができる。

4.2 語句に関する機能

公理によって意味定義される語句を扱うための機能を述べる。なお、ここでいう語句とは、語句関連グラフにおいて頂点にラベル付するもの、あるいはすでにされているものを指す。

目的とする語句は、新たに登録する場合など一部を除き、語句指定子を用いて指定する。語句指定子には、以下の3通りのものがある。

1) 語句を表す文字列

登録されている語句そのもので指定する、あるいは特殊記号を用いて、特定の文字または単語を含む語句として指定する方法である。この指定方法を用いれば、一度に複数の語句を指定することができる。

2) 語句識別番号

全ての語句は登録される際に、それぞれ固有の識別番号が付けられる。その識別番号は、正の整数値であり、その語句が消去されるまで、いかなる処理を行っても変化しない。

ユーザは、この識別番号によって目的とする語句を指定することができる。

識別番号は、語句を登録したときや語句を表示させたときに知ることができる。

3) 語句ポインタ

本システムは、特定の語句を頻繁に指定するような場合に備えて、4個の語句ポインタを用意している。

ユーザは、語句ポインタに任意の語句を記憶させることができ、以後ポインタの番号によってその語句を指定することができる。語句ポインタの番号は0から3までの整数値で、値の前に 'p' を付けることで、語句ポインタの番号であることを示す。

語句ポインタに語句を記憶させるにはコマンド 'setp' を用いる。

setp 語句ポインタ番号 語句指定子

この場合の語句ポインタ番号には 'p' は付けない。

語句ポインタに記憶されている語句は常に画面の上端に表示されているので、ユーザは各語句ポインタの内容を覚えておく必要はない。

なお、全てのコマンドにおいて、語句指定子は省略することができ、その場合には語句ポインタの0番、すなわち 'p0' なる指定がなされたものとして処理される。た

だし、その際には0番の語句ポインタには語句が記憶されていなければならない。

新たな語句が登録された場合、あるいは語句を表示させる際に3)の指定方法を用いた場合に、その新たな語句、あるいは指定された語句が自動的に0番の語句ポインタに記憶される。特に、表示の際3)の指定方法が用いられた時には、指定された語句ポインタと0番の語句ポインタの内容が交換されるので、それまで0番に記憶されていた語句が失われることもない。

一度に複数個の語句を指定できるのは、1)の方法による場合のみである。

a) 語句の登録

mkp

コマンド'mkp'により新たな語句を登録できる。登録する語句の種類(名詞句、動詞句など)や語句は、システムと対話的に指示する。登録の際には、その語句は構文解析され、その構文木が格納される。登録に際して、その語句には識別番号が付けられ、その識別番号が画面に表示される。また、その語句は0番の語句ポインタに記憶される。

語句は項目名と異なり、同じ語句が複数個存在することもできる。

b) 語句の削除

rmp [オプション] 語句指定子

項目の削除と同様、オプション'-i'をつけることができる。

c) 語句の変更

chp 語句指定子 新語句

語句指定子が複数個の語句を表す場合は、それぞれの語句に対して変更するか否かの確認を行なう。

d) 関連語句の宣言

rel 語句指定子₁-語句指定子₂ 関係名

指定された語句₁から指定された語句₂へ、指定された関係を付ける。

語句間の関係を宣言するためには、その関係名はすでに登録されていなければならない。関係名を登録する時は、同時にその逆関係名も登録する。関係名を登録、変更、または削除するためのコマンドは、

mkrelname 関係名 逆関係名

chrelname 旧関係名 新関係名

rmrelname 関係名

である。

e) 語句を意味定義する公理の宣言

def 語句指定子-公理識別番号

指定された語句の意味定義に、指定された公理が用いられることを宣言するものである。一度宣言されたものを取り消すには、コマンド'undef'を用いる。

f) 語句の表示

prp [オプション] 語句指定子

オプション

-c 表示前にウィンドウを消去する。

-r その語句と関連のある語句も表示する。

-a その語句を意味定義する公理も表示する。

指定された語句に関する情報を表示する。オプションにより、関連語句や公理も表示させることができる。ただし、ここで表示される公理は、自然語文の形のみである。

これにより、特定の語句を意味定義する公理文を検索することができる。

4.3 公理に関する機能

公理は項目や語句とは異なり、それを指定する方法は公理識別番号による方法のみである。公理識別番号は、公理が登録された時に付けられる固有の整数値であり、語句識別番号同様、その公理が削除されるまで変化しない。

a) 公理の登録

mkax 公理文(自然語)

与えられた公理文(自然語で書かれたもの)を構文解析して論理式に変換後、自然語文と変換後の論理式との構文木を辞書に蓄積する。また、その登録者、登録日時も自動的に記録される。

b) 公理の削除

rmax 公理識別番号

c) 公理の変更

chax 公理識別番号 新公理文(自然語)

指定された識別番号の公理文を新たな公理文に変更する。新たな公理文は、登録時同様、構文解析後論理式に変換されて記録される。また、その変更者、変更日時も自動的に記録される。

d) 公理の属性

addatr 公理識別番号 属性名

公理に何らかの属性を付けるためには、その属性名をあらかじめ登録しておかなければならない。

属性名を登録、消去するコマンドは、

mkaturname 属性名

rmaturname 属性名

である。

e) 公理の表示

書式1: prax [オプション] 公理識別番号

書式2: prax [オプション] all

書式3: prax [オプション]

オプション

-c 表示前にウィンドウを消去する。

-l 論理式の公理文も表示する。

-a その公理文に付けられている属性も表示する。

-e 登録者、登録日時も表示する。

-m 変更者、変更日時も表示する。

これは、特定の公理文に関する情報を知りたいときに使用するものである。書式1では指定された識別番号を持つ公理を表示し、書式2では全公理を表示する。書式3の場合、語句ポインタの0番に記憶されている語句を意味定義している公理を表示する。オプションにより、その公理文に関する各種の情報をすることもできる。

4.4 項目による公理の検索

上記の機能を用いて項目から語句を検索することや、語句を意味定義する公理を検索することができる。

本システムは、さらに、項目から語句と公理とを検索できる機能を持っている。

pr 項目指定子

コマンド 'pr' は、指定された項目から検索できる語句の意味定義となる公理を表示するものである。その項目から検索できる語句が複数個あるときは、優先順位の高いものから順に、その語句と公理とを表示してユーザの確認を求め、確認された語句は、最も優先順位の高い位置に置き換えられる。

項目指定子で指定される語句が複数個あるときは、語句や公理を表示する前に、その項目がユーザの意図したものか否かの確認をとる。

4.5 特殊記号について

本システムでは、項目や語句を指定する際にワイルドカード文字などの特殊記号を使用できる。これにより、名前の一部を省略して指定したり、特定の単語を含むものを選びだすことが可能である。特殊記号として使用できる文字の種類やそれが意味するものは表1の通りである。

表1 特殊記号の種類とその意味

記号	その記号の意味するもの
*	空を含む任意の文字列
?	任意の1文字
^	空白、または、文字列の先頭、末尾
[文字 ₁ —文字 ₂]	文字コードが文字 ₁ 以上文字 ₂ 以下であるもの
\文字	その文字自身

表2 特殊記号の使用例

a*	第1文字がaであるもの
[b-d]*	第1文字がb, c, またはdであるもの
of	文字列ofを含むもの
^of^	単語ofを含むもの
***	文字*を含むもの

る。

5. その他の機能

ここでは、ユーザ・インタフェース向上のために用意されている機能を紹介する。

5.1 オプションについて

4で述べたように、表示の際、各種のオプションを使用することができる。ここでは、それらのオプションに関する機能を述べる。

a) 全ての情報を知りたい場合

コマンドのオプションを全て指定する方法もあるが、それに代わる方法として、オプション '-A' を指定することができる。

b) オプションのデフォルト化

特定のオプションを頻繁に使うような場合、それらのオプションをコマンドのデフォルトとすることができれば、毎回オプションを指定しなければならないという煩しさが解決できる。

本システムでは、コマンド 'dft' を用いてデフォルトとなるオプションをユーザが指定することができる。

dft コマンド名 オプション

これにより、以後は常にデフォルトオプションが指定されているとして各コマンドが実行される。また、

dft コマンド名

により、そのコマンドのデフォルトオプションを知ることができる。

5.2 help機能

コマンド'help'により、本システムの全コマンドとその簡単な書式説明を得ることができる。さらに、

help コマンド名

により、機能などそのコマンドに関する細かな情報を知ることができる。

5.3 alias機能

alias 登録コマンド名 コマンド [オプション]

コマンド名の長いものを頻繁に使用する場合は、先に述べた'dft'機能とは別にコマンドに特定のオプションを付けてそれを異なるコマンドにしておきたい場合などに用いる機能である。

5.4 コマンド・ファイル

一定のコマンド列を1つのコマンドで処理できるようにするための機能である。

コマンド列をあるファイルに書き込んでおけば、そのファイル名をコマンド名として使用できる。さらに、ファイル名が'.start'であれば、そのコマンド・ファイルは本システム起動時に自動的に実行される。

6. あとがき

以上、分野がプログラムの仕様記述に用いるものと限定された自然語の処理系を、辞書管理の機能を中心に紹介した。本システムはUNIX上に構築されており、構文解析部はProlog、その他の部分はCにより記述されている。その大きさは、コマンド・インタプリタは約2000行、構文解析部は約300行(Prolog)、論理式への変換部は約2500行、辞書管理部は単語辞書管理部が約500行、公理辞書管理部が約2000行である。

本システムの今後の改良点や拡張点として以下のことが挙げられる。

例えば、語句の意味定義となる公理を登録する際、現在は自然語で書かれたものを論理式に変換する方法をとっているが、公理によっては直接論理式で記述する方が容易な場合もあり、そのためには与えられた論理式を構文解析する機能も必要である。

また、現在のシステムでは辞書は1通りしか存在しないが、辞書を、プログラム化する問題の分野別、または個人別などに分割して、必要なものだけを使用できるようにすれば、語句の意味を検索する際にも目的とするものをより効率的に検索することができるであろう。

さらに、自然語文や論理式の構文木表示形式、語句関

連状況の表示形式などユーザ・インタフェースに関しても検討が必要である。

【謝辞】

日頃、御指導を頂く大阪大学基礎工学部 嵩忠雄教授に深謝致します。また、本システム構築の一部を担当して頂いた本学 御井敬氏に感謝致します。

【参考文献】

- (1) K. Inoue, H. Seki, K. Taniguchi and T. Kasami: "Compiling and Optimizing Methods for the Functional Language ASL/F", Science of Computer Programming, Vol. 7, No. 3, pp. 297-312 (1986).
- (2) 東野, 工藤, 縄田, 杉山, 谷口: "代数的仕様検証支援系及びそれを用いた検証例", 信学論(D), Vol. J67-D, No. 4, pp. 472-479 (1984).
- (3) 安松, 杉山, 鳥居: "抽象的順序機械からASL/Fへの変換システムについて", 情処研報, 86-SW-46-4 (1986-02).
- (4) 関, 並河, 藤井, 嵩: "自然語によるプログラム仕様の形式的意味定義について", 信学技報, COMP86-2 (1986-05).
- (5) 関, 並河, 藤井, 嵩: "自然語によるプログラム仕様の形式的意味定義 - 自然語による仕様から代数的仕様への変換 -", 情処研報, 87-SW-52-7 (1987-02).