数学的構造を反映させた定理証明系の試み

4J-1

廣瀬健[1]，桔梗宏孝[1]，山田眞市[1,2]

[1]早稲田大学理工学部　[2]日本ユニシス株式会社

1．はじめに

既存の証明論に基づいて構築された自動証明系は，一般に大きな知識ベースを必用とし，また式が大きくなるに連れて組み合わせ論的な爆発により手に負えなくなりやすい．実際の理論はある種の領域を意識して作られており，多くの場合ある種の群，環，束など構造のはっきりわかっているより抽象的な構造と同型や準同型で関係付けられている．この同型，準同型に付随して，1つの領域上の元，関数，関係を他の領域上で表現する表現定理がある．実際の数学的思考に際しては，この表現定理を使ってよりやさしい領域に移して問題を考えることが多い．ここでは，理論ごとにこの表現定理をメタ知識として用意しておき，よりやさしい理論の上に翻訳して問題を解き，元に戻すという方法を組み込んだ自動証明系を紹介し，この方法の有効性を述べる．

2．証明手続き

次のような手続きを自動証明系に組み込む．

a．問題が与えられると，それをより抽象的な理論の問題にメタ知識として用意された表現定理を使って変換する．

b．変換された方の理論で，その問題を解く．一般には，その問題が完全に解けるわけではなく，部分的な解となる．

c．b．で得た解をもとの理論に変換する．

一般には，この操作を複数の抽象的理論との間で繰り返す．

3．　例

上の方法がうまくいく代表的な例をあげる．

常微分方程式

正の実数上の連続関数の全体は，各点ごとの加法と畳み込みにより整域をなす．その商体をDとする．線型の常微分方程式は，Dの中である有理式として解ける．それを部分分数展開すれば，もとの関数に戻せる（ミクシンスキの演算子法）．

偏微分方程式

線型の偏微分方程式は超関数の空間へ移すと解ける（フーリエ変換，ラプラス変換）．超関数の空間は，ある関数空間の双対空間で，ほとんどの計算が内積の計算で出来る．また，ある種の条件のもとで，解をもとに戻せる．

積分論

実数上の積分は，ベクトル束上のノルムである．積分に関する多くの定理は，束の上の計算でできる．

4．試作システム

現在，常微分方程式を解くシステムと超関数に関する証明系の一部がMicro-Vax IIの上にQuintus Prologで実現されている．これらのシステムの結果を以下に報告する．

常微分方程式については，整数係数の常微分方程式が解けるものが出来ている．図1，2がその出力の例である．図1は，一般解を求める例であり，図2は，ある初期条件のもとでの解を求める例である．このシステムでのメタ知識は，関数の微分のミクシンスキの微分演算子sを用いた表現と三角関数や指数関数，多項式などのsを用いた表現である．これらは，比較的小さい表として用意しておくことができる．逆の変換のメタ知識として，sの有理式は，部分分数展開をして表をつかうということが用意されている．常微分方程式をオペレータの空間に翻訳すると，解は単に割り算をすればもとまり，sの有理式になる．よって，部分分数展開が出来れば良いことになる．オペレータの空間は体であるから，これには既存の数式処理が使える．このシステムでは，因数分解の数式処理系*)だけを他から借り，残りはすべてわれわれで構築した．この部分は1000行ほどである．図1，2のような例の場合，10数秒で解が求まる．偏微分方程式についても同じ様なシステムが出来る．こちらの場合，オペレータの空間が体であったので既存の数式処理がすぐに使えたのにたいし，超関数の空間はそうではないので，超関

Theorem Proving System Reflecting Mathematical Structures

Ken HIROSE[1], Hirotaka KIKYO[1], Shinichi YAMADA[1,2]

[1]Waseda University, [2]Nihon Unisys Ltd.

*）富士通国際情報社会科学研究所で開発中

```
 (4)
x    - 2 x''' + 2 x'' - 2 x' + x = f

General Solution.

 4      3      2
(s  - 2 s  + 2 s  - 2 s + 1) x  =  W + f

x =  a1*s/(s^2+1^2) + a2/(s^2+1^2) + a3/(s-1)^2 + a4/(s-1)

   + (1/2*s/(s^2+1^2) + 1/2/(s-1)^2 + -1/2/(s-1))*(f)

  = a1 cos(t) + b2 sin(t) + a3 t exp(t) + a4 exp(t)

   + (1/2 cos(t) + 1/2 t exp(t) - 1/2 exp(t))*f
```

図１　常微分方程式の一般解

```
 (8)      (6)
x    + 2 x    - 2 x'' - x = 0

x(0) = 0, x'(0) = 2, x''(0) = 0

               (4)          (5)
x'''(0) = 2, x    (0) = 0, x    (0) = -1

 (6)          (7)
x    (0) = 0, x    (0) = 11

 8      6      2                  6      4      2
(s  + 2 s  - 2 s  - 1) x  =  2 s  + 6 s  + 3 s  + 5

x =  1/(s-1) + -1/(s+1) + -3/(s^2+1^2)^3

  = exp(t) - exp(-t) - 3 sin(t)*sin(t)*sin(t)
```

図２　常微分方程式の初期値問題

```
formula=[d(heaviside),f].


  [d(heaviside),f]
= -[heaviside,d(f)]      by ii_2_6
= -[$,[0,infinite],d(f)]      by definition
= - (-f(0))      by definition
= f(0)      by definition

Hence, [d(heaviside),f]=f(0).

c/s ?c.
formula=[delta,f].


  [delta,f]
= f(0)      by ii_1_22

Hence, [delta,f]=f(0).
```

図３　ヘビサイド関数とデルタ関数

数の理論の自動証明系から始めた．超関数の計算は，ほとんど双線型形式（内積）の計算で出来る．図３は，ヘビサイド関数の微分を計算したものである．これらは，簡単な項書換え規則によって少ない手数で求められている．メタ知識は，超関数の微分の内積による表現などである．

この方法は，複雑な空間から抽象的な空間を抽出しておいて，与えられた問題をなるべくきれいにやるという方法であり，得られた計算過程や証明過程はかなり人間にとっても自然なものになっている．また，抽象的な部分を独立に作れるので，システムとしてもコンパクトですっきりしたものができ，開発にもあまり時間がかかっていない．因数分解など計算量の大きい部分もあるが，次数の低い実際にでてくる微分方程式などでは問題にならない．一般にこれらは手計算でもできるが，変換だけでも大変な場合が多く，計算機でやることが有効な方法になっている．さらに，うまく行かないところを対話的に処理をすることにより，問題の本質的に難しい部分を抽出することに役立つと思われる．

5．おわりに

現在，上のシステムを拡張して，線型偏微分方程式，連立常微分方程式，例３の積分論なども扱えるシステムにする作業を進めている．これらも開発に余り時間がかからないと思われる．抽象的な理論の証明システムを組み合わせることにより，いろいろな分野の実際に役立つコンパクトなシステムが出来るはずである．

参考文献

[1] K. Hirose, H. Kikyo et al, Use of Meta-Knowledge on Mathematical Structures in Theorem Proving, 3rd Asian Conf. in Math. Logic, 1987.

[2] H. Kikyo, An Implementation of a Theorem Proving System Based on Proof Procedures Using Knowledge on Mathematical Structures, Bull. of the Contre for Informatics, Vol 7 (1988), Waseda Univ.