

HTML をもちいた定理の自動証明システムの実現

4Aa-8

西田雅彦 高津智仁 辰巳丈夫 笈捷彦  
早稲田大学理工学部

1 はじめに

定理の自動証明システムを WWW 上で実現する。

一般にハイパーテキストのビューワーと認識されている WWW ブラウザを一つの環境ととらえ、その上に論理体系 CLC に基づく証明システムを構築する。

WWW ブラウザとサーバ上の証明プログラムの間で、命令とデータのやりとりを繰り返しながら証明を進める。証明プログラムはサーバ上で実行されるので、ユーザーはただ WWW ブラウザを用意するだけでシステムを利用することができる。つまりコンピュータの機種や OS の種類によらない普遍的な環境を提供することができた。

<http://prover.kake.info.waseda.ac.jp/index-j.html/>

2 HTML を用いた証明システム

2.1 システムの全体

今回我々が作成したシステムは、Gentzen 流のシーケント計算に基づく体系 CLC を用いている。

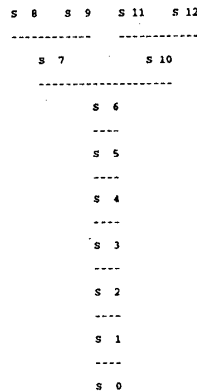
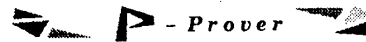
このシステムで、ユーザーがシステムに与えることのできる命令は、次の通りである。

1. 証明したい論理式を入力する。(最初の一回だけ。)
2. 指定したシーケントを自動で非アラール分解する。(分解できなくなるまで。)
3. 指定したシーケントの指定した場所をアラール分解する。
4. アラール変数の候補をあげ、それらをイニシャルシーケントではないシーケントすべてに代入し、評価する。

まず最初に、命令 1 で証明したい論理式を入力する。次に、証明の過程にある証明木とデータを見て、命令

An implementation of automated theorem prover on WWW server.

NISHIDA Masahiko, TAKATSU Tomohito,  
TATSUMI Takeo, KAKEHI Katsuhiko  
School of science and engineering, Waseda university.



```

S0: ! any x(A(x) imply B(x)) and any x(B(x) imply C(x)) imply any x(A(x) imply C(x))
S1: any x(A(x) imply B(x)) and any x(B(x) imply C(x)) ! any x(A(x) imply C(x))
S2: any x(A(x) imply B(x)),any x(B(x) imply C(x)) ! any x(A(x) imply C(x))
S3: any x(A(x) imply B(x)),any x(B(x) imply C(x)) ! A(eigen0) imply C(eigen0)
S4: A(eigen0),any x(A(x) imply B(x)),any x(B(x) imply C(x)) ! C(eigen0)
S5: A(eigen0),any x(A(x) imply B(x)),A(alah0) imply B(alah0),any x(B(x) imply C(x)) ! C(eigen0)
S6: A(eigen0),any x(A(x) imply B(x)),A(alah0) imply B(alah0),any x(B(x) imply C(x)),B(alah1) imply C(alah1) ! C(eigen0)
S7: A(eigen0),any x(A(x) imply B(x)),any x(B(x) imply C(x)),B(alah1) imply C(alah1) ! A(alah0),C(eigen0)
S8: A(eigen0),any x(A(x) imply B(x)),any x(B(x) imply C(x)) ! B(alah1),A(alah0),C(eigen0)
S9: A(eigen0),any x(A(x) imply B(x)),any x(B(x) imply C(x)),C(alah1) ! A(alah0),C(eigen0)
    
```

図 1: サンプル

2,3,4 をユーザーが指示する。命令を一回出すごとに、古い HTML ソースが、新しい HTML ソースに置き換わる。これを繰り返すことにより、証明を進める。

3 システムの特徴

主プログラムはユーザーの一回の命令の度に実行を終了し途中までの証明を破棄するので、効率的に証明を進めるためには途中までの証明を保存しなければならない。ファイルで保存することも考えられたが、WWW で公開するということから、不特定多数の人が同時に使うという特殊性を考慮しなければならない。

そこで、HTML ソースに隠し属性で、途中までの証明を保持することにした。この方法を採用することで、複

数の人が同時に使用することができ、ファイルでディスクがあふれる心配もない。

また、途中までの証明を HTML ソースに隠し属性で保持しているので、それを各自が保存しておけば、次回はそのファイルを用いて前回の続きから証明を続けることができる。また、完成した証明を別の用途に利用することもできる。

#### 4 システムの問題点

いろいろな論理式を用いて調べてみると、今回作成したシステムにはいくつかの問題があることがわかった。

1. 命令を出すたびにすべてのデータがネットワークを通して転送され、ネットワークに負担がかかる。
2. 命令を途中で取り消すと、サーバ上のプロセスが終了せずに残る。
3. アラー変数を代入評価するプログラムの実行が特に遅い。

問題点1と2は、HTML プロトコルの設計上の問題である。

問題点1の原因は、HTML ソースにすべてのデータを保持する方法をとったことによる。しかしいろいろな論理式で調べてみると、待ち時間の多くは主プログラムの実行時間であり、データ転送の時間はあまり重要ではないということがわかった。

問題点2は、この種のサーバを公開するにあたり問題となる。

WWW では、ユーザの要求に対して反応が遅いと、ユーザーはブラウザのストップボタンを押し実行をやめようとするが、現在の実装ではサーバ側にはプロセスが残る。このプロセスが、サーバにたくさん溜ると、システムの反応がさらに遅くなり、サーバそのものが停止してしまう。

多くの人に開放するという目的を達成できるかどうかは、この問題にどのように対応するかにかかっている。

問題点3は、問題点2と関係がある。

分解命令に比べ、代入評価命令はかなりの時間がかかる。問題点2で残ってしまうプロセスの多くも代入評価のプロセスである。

これはアラー変数またはアラー変数の候補が増えるにつれ、代入評価にかかる時間が爆発的に大きくなるから

である。

以上の問題に対しに、つぎの方法で改良を行なうことを予定している。

1. より速い代入評価プログラムを作る。
2. 残留プロセスを消去する機能を持たせる。
3. 同時にアクセスできる人数を制限する。

WWW 上で多くの人に向けて公開する目的を考えると、対応策3をとる場合にはなるべく制限される人数を多めにとるべきである。

まず対応策1,2を採用して、改良を行なう予定である。

#### 5 おわりに

今回の研究の主題は、普遍的な環境を広く一般に提供することにある。

できるだけ多くのブラウザに対応するために、標準的な HTML コードを使用することを念頭においた。しかし、ユーザーインターフェースはユーザーの用意するブラウザに依存する部分が多い。

たとえばこのシステムでは、試行錯誤して証明を進めるために必要な「証明の後退」を、ブラウザの「Back」に頼っているが、ブラウザによっては「Back」がうまく働かない場合がある。また、どのくらいの大きさの証明木を表示できるかもブラウザに依存している。

ブラウザの普遍性に注目し考えられたシステムなので、多くのブラウザに対応することが必要である。

#### 参考文献

- [1] 加藤雅英. 論理体系 CLC を用いた定理証明支援システムの設計-ユーザインターフェイスを考えたアラー分解とアラー変数への代入方法との提案-. 早稲田大学理工学部情報学科 1994 年度卒業論文,1995.
- [2] 柴山能成. 定理証明支援システムにおけるユーザー・インターフェイス. 早稲田大学理工学部情報学科 1994 年度卒業論文,1995.
- [3] 林久志. 論理体系 CLC を使った定理証明支援システムの設計及び実現. 早稲田大学理工学部情報学科 1994 年度卒業論文,1995.