非古典論理に対するシーケント計算における効率的な証明探索法

丸山 哲 (010113)

北陸先端科学技術大学院大学 情報科学研究科

2002 年 5 月 15 日

キーワード: sequent 計算, 直観主義論理, 定理自動証明, Standard ML.

1 研究目的

非古典論理に対する定理の自動証明はさまざまなアプローチがある。本研究では cut のないシーケント計算を用いた証明探索について比較、検討をおこない、さらに直観主義論理の体系に対して、 Standard ML を用いて計算機への実装をおこなった。

2 証明探索とその体系

2.1 Wang の体系

良く知られているように cut 除去定理がなりたつ体系の多くにおいては、その subformula property を用いることにより決定可能性を導くことができる。しかしながら、決定可能性を示すために用いられる通常の決定アルゴリズムは LK の場合でさえもきわめて効率が悪い。それは、証明の探索を実行している際に新たに得られた式が、これまでの探索ですでに得られているかどうかをチェックする必要があるからである。このチェックはループチェックと呼ばれるが、ループチェックはメモリと時間の両方を必要とするため効率を下げる大きな要因となる。

古典論理に対しては体系 LK を変形して得られる Wang の体系が知られている。Wang の体系では見かけ上 contraction の規則がなくなっているためにループチェックを必要としない。さらに Wang の体系は上式と下式の証明可能性が同値になっているため、全く機械的な証明探索が可能になり試行錯誤を行なう必要がない。そのため、きわめて効率のよい証明探索が可能になる。

2.2 Dyckhoff と Hudelmaier の体系

この Wang の体系と同様のアイディアに基づく直観主義論理の体系に合わせたものに Dragalin の体系がある (図 1 参照)。しかし、この体系は $(\supset\!\rightarrow^*)$ の規則においては、左上式が下式よりも簡単になっているとは限らないために停止性が保証されず、やはりループチェックをする必要が生じてくる。

この問題の解決するために 1990 年代始めに Dyckhoff と Hudelmaier は独立に新たな体系を与えた。その体系は Dragalin の体系の $(\supset\!\rightarrow^*)$ を図に示すような 4 つの規則に置き換えたものである。

$$\frac{}{A,\Gamma \rightarrow A,\Delta}\ (action^*) \qquad \frac{}{f,\Gamma \rightarrow \Delta}\ (f\ \rightarrow^*)$$

$$\frac{A,B,\Gamma \rightarrow \Delta}{A \wedge B,\Gamma \rightarrow \Delta}\ (\wedge\ \rightarrow^*) \qquad \frac{\Gamma \rightarrow A,\Delta \quad \Gamma \rightarrow B,\Delta}{\Gamma \rightarrow A \wedge B,\Delta}\ (\rightarrow \wedge^*)$$

$$\frac{A,\Gamma \rightarrow \Delta \quad B,\Gamma \rightarrow \Delta}{A \vee B,\Gamma \rightarrow \Delta}\ (\vee\ \rightarrow^*) \qquad \frac{\Gamma \rightarrow A,B,\Delta}{\Gamma \rightarrow A \vee B,\Delta}\ (\rightarrow \vee^*)$$

$$\frac{A \supset B,\Gamma \rightarrow A \quad B,\Gamma \rightarrow \Delta}{A \supset B,\Gamma \rightarrow \Delta}\ (\supset\!\rightarrow^*) \qquad \frac{A,\Gamma \rightarrow B}{\Gamma \rightarrow A \supset B,\Delta}\ (\rightarrow\!\supset^*)$$

図 1: Dragalin の体系

$$\frac{B,A,\Gamma \rightarrow \Delta}{A \supset B,A,\Gamma \rightarrow \Delta}\ (\supset\!\rightarrow^*_1)$$

$$\frac{C \supset (D \supset B),\Gamma \rightarrow \Delta}{(C \wedge D) \supset B,\Gamma \rightarrow \Delta}\ (\supset\!\rightarrow^*_2)$$

$$\frac{C \supset B,D \supset B,\Gamma \rightarrow \Delta}{(C \vee D) \supset B,\Gamma \rightarrow \Delta}\ (\supset\!\rightarrow^*_3)$$

$$\frac{D \supset B,\Gamma \rightarrow C \supset D \quad B,\Gamma \rightarrow \Delta}{(C \supset D) \supset B,\Gamma \rightarrow \Delta}\ (\supset\!\rightarrow^*_4)$$

図 2: Dyckhoff と Hudelmaier の体系

この体系では部分的に試行錯誤 (いくつかの選択の可能性) は存在するが従来の方法に比べてはるかに効率がよいものになっている。

この Dragalin と Dyckhoff と Hudelmaier の体系を用いた証明探索法の有効性を確認するため、この研究ではこのアルゴリズムを計算機に実装し、いくつかの実験をおこなった。

3　実装について

3.1　入力方法

入力については Standard ML より下のように行う。

- 命題変数は ‘Prop’ という宣言を先頭に入れる（例：Prop ”A”）
- 論理式 $A \wedge B$ は Land (Prop ”A”, Prop ”B”) と入力する
- 論理式 $A \vee B$ は Lor (Prop ”A”, Prop ”B”) と入力する
- 論理式 $A \supset B$ は Limp (Prop ”A”, Prop ”B”) と入力する
- 複数の論理式や命題変数を入力する場合には、それぞれの論理式の間にコンマ ‘,’ を入力する（例: $A \wedge B, C \vee D$ の場合、‘Land (Prop ”A”, Prop ”B”), Lor (Prop ”C”, Prop ”D”)’ と入力）

この他に証明を行わせる関数などを補う必要があるため、実際の入力形式は以下のようになる。

```
main (node ([論理式左辺], ([], [])), node ([論理式右辺], ([], [])));
```

論理式を入力すると証明可能な場合には証明図を出力することになる。しかし、証明不可能な場合には Standard ML の例外が発生するので、 ‘NO_PRINCIPAL’ とだけ表示される。

4　まとめ

本研究では効率のよいシーケント計算により体系を基礎とする prover の実装を行なった。実際、直観主義論理の Dyckhoff と Hudelmaier の体系における prover の実装を行うことができその有効性を確認することができた。