P-1

Isabelle による並列処理を行う自動証明システムの構築
Emacs Lisp, PostgreSQL を用いて
Construction of Automated Reasoning System Performing Parallel Processing by Isabelle Using Emacs Lisp, PostgreSQL

○馬場彰太郎[1]，小林英恒[2]
*Shotaro Baba[1], Hidetsune Kobayashi[2]

Abstract: Isabelle is an automated reasoning program for the proofs of mathematical formulas, and software verifications. Kobayashi developed the automated reasoning program using Emacs Lisp and PostgreSQL. However, it is not equipped with the functionality to execute automated reasoning by parallel processing. This paper explains the outline of the parallel processing program for automated reasoning.

１．概要

Isabelle は定理証明システム[1]と呼ばれるプログラム言語で，数学の定理証明，プログラムの動作検証，暗号の安全性の検証に主に用いられる．小林[2]は Isabelle およびプログラミング言語である Emacs Lisp およびデータベース管理システム PostgreSQL を用いて自動証明のプログラムを作成したが，並列処理は用いられていない．本論文では，その並列処理法についての概要および課題について述べる．

２．Emacs Lisp、PostgreSQL、および Isabelle について

Emacs Lisp は，テキストエディタ Emacs に特化した Lisp 言語である．また，PostgreSQL はデータベース管理システム[3]構築に用いられる言語である．その一方，Isabelle は数学における証明の検証，プログラム検証，暗号の安全性の検証に主に用いられるプログラミング言語[1]であり，これまで主に，素数定理の証明，Java 言語の検証，暗号の安全性の検証などに用いられてきた．

以下，Isabelle における数学の定理証明の検証過程について具体的な例を用いて説明する．

```
lemma "⟦A; A ⟶ B; B ⟶ C⟧" ⟹ C
apply (drule mp, assumption)
apply (drule mp, assumption+)
done
```

まず，Isabelle は与えられた命題に対し証明するべき命題を数学の証明に則って提示する．次に，提示された命題に応じてプログラムを書き込まれ，それに沿って証明は進む．また，証明すべき命題は順次 Isabelle により再度表示される．

上記のように Isabelle を用いることで，証明の過程に類似した記述の仕方でプログラムを作成できる．以下，Isabelle で用いられる３つのコマンドについて説明する：lemma は証明する命題をプログラムに与えるコマンドで，""の中にその命題を書き込む．apply は命題に証明を与えるコマンドで，( )内にそのコマンドを書き込む．done は証明の終了を Isabelle に伝えるコマンドで，Isabelle において証明する必要がある命題が残っていた場合はエラーとなる．

３．自動証明プログラム[2]の概要

自動証明プログラムとは Isabelle において証明する必要のある命題から，Isabelle に用いられるプログラム文を自動的に生成し証明を進行するようなプログラムである．プログラムは Emacs Lisp および PostgreSQL で作成される．以下に自動証明の概要を示す：

1. Isabelle において証明する必要がある命題が提示されたとき，それを取り出す．
2. 取り出した命題を Emacs Lisp のリストに変換する．
3. 2.で変換したリストから Isabelle に用いられるプログラム文字列を作成する．
4. 作成した文字列をプログラム文に挿入する．
5. 証明が終了するまで 1~4 を繰り返す．

1~4 までの具体例を以下に示す：　まず，証明すべき命題が以下であったとする：

```
lemma "⟦A; A ⟶ B; B ⟶ C⟧ ⟹ C"
```

ここで、命題を Lisp 言語のリストに変換する：

```
((((A)) (lrarS (A) (B)) (lrarS (B) (C))) (C))
```

次に，証明のための文字列を次行に入れる：

```
lemma "⟦A; A ⟶ B⟧ ⟹ B"
```

１：日大理工・院（前）・数学　２：計算論理研究所

apply (rule mp)

それ以降 Isabelle が終了判定を行うまで，1 から 4 の動作が繰り返される．

４．並列処理の概要

並列処理の方法について以下に示す：

1. Isabelle において与えられた命題を Lisp のリストに変換する．
2. ホストがその命題に適用できる定理の種類と数を確かめる．
3. ホストに残った処理データをクライアントが読み込み，自動証明を行う．
4. 残りの処理を行ったクライアントの自動証明が終了したとき，ホストに処理終了を伝え，クライアントの処理を終了させる．

上記中，ホストとは自動証明の最初の段階を行うコンピュータを意味し，クライアントとはホストに残った処理データを取ってきて自動証明処理を行うコンピュータのことである．

次に，並列処理について説明する：ホスト・クライアント間の通信には PostgreSQL と TFTP を用いる．処理過程の概要は Figure 1 のようになる．Isabelle の文字列を読み込んだとき，それを言語 Emacs Lisp のリストに変換する．ここで，H をホストとし，読み込まれた命題における証明で適用できる定理が k 個存在したとする．ホストは各々の定理に対し，$T_1,T_2,...,T_k$ と対応付ける．次に，ホストは $T_1,T_2,...,T_k$ の定理を適用する場合の処理をそれぞれ $rt,rt_1,rt_2,...,rt_{k-1}$ と定義する．ここで，H は rt を処理する．一方，クライアント $C_1,C_2,...,C_{k-1}$ はホストに残された処理を行う．次に，C_1 は rt_1，C_2 は rt_2，...，C_{k-1} は rt_{k-1} を処理する．ここで，$C_1,C_2,...,C_{k-1}$ の自動証明の方法は前節と同じである．最後に，ある定理 T_m を適用する証明処理が他のクライアントよりも早く終了した場合，そのクライアントがホストに終了したという情報を送り，受け取ったホストはすべてのクライアントの処理を終了させる．また，ホストは定理 T_m を Isabelle に適用するため，並列処理の際に読み込んだ Isabelle のプログラム文に作成されたプログラム文を挿入し，全ての処理が終了する．

５．展望と課題

並列処理を行うことにより，速い速度の自動証明が可能となることが期待される．学術講演会当日では，並列処理プログラムとそれを実行した場合の考察について発表を行う予定である．

６．参考文献

[1] Tobias Nipkow, Lawrence C. Paulson, Markus Wenzel, “Tutorial for Isabelle,” 2012.
[2] 小林英恒, ar07Sep12.el, 2012.
[3] PostreSQL Global Development Group, “PostgreSQL 9.1.3 Documentation,” 2012.

Figure 1.Outline of the parallel processing for automated reasoning