



Design and implementation of linear logic programming languages

番原, 睦則

(Degree)

博士 (工学)

(Date of Degree)

2002-09-20

(Date of Publication)

2007-08-09

(Resource Type)

doctoral thesis

(Report Number)

乙2637

(URL)

<https://hdl.handle.net/20.500.14094/D2002637>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



【 2 9 6 】

氏 名 ・ (本 籍) 番 原 睦 則 (岡 山 県)

博士の専攻分野の名称 博士 (工学)

学 位 記 番 号 博ろ第252号

学位授与の 要 件 学位規則第 4 条第 2 項該当

学位授与の 日 付 平成 1 4 年 9 月 2 0 日

【 学位論文題目 】

Design and Implementation of Linear Logic Programming Languages

(線形論理型言語の設計と実装)

審 査 委 員

主 査 教 授 角 田 讓

教 授 金 田 悠紀夫

教 授 新 井 敏 康

助教授 田 村 直 之

本論文では、線形論理および時相線形論理に基づく論理型プログラミング言語のための新しい実装方式を提案した。特に、以下の点に新規性がある。

- 線形論理型言語のコンパイラ処理系：
コンパイラ処理系のための抽象機械および命令セットの提案。
- 線形論理型言語から Java へのトランスレータ処理系：
Java へのトランスレート方式の提案、ランタイム・システムの設計。
- 時相線形論理型言語のコンパイラ処理系：
時相線形論理型言語の設計、計算モデル、コンパイラ処理系のための抽象機械および命令セットの提案。

線形論理は 1987 年にフランス人論理学者 J.-Y. Girard が提案した新しい論理体系である。線形論理は“リソースを意識した論理”と呼ばれ、従来の論理(古典論理, 直観主義論理など)の拡張になっている。線形論理は情報科学の分野に広く応用されており、従来の論理では困難であった問題に対して、新しい解決法を与えている。その豊かな表現力は、ベトリネット, π 計算などの並行計算モデルを自然に表現できることから示される。また近年、セキュリティ・プロトコルの操作的意味論, リアルタイム・システムの仕様記述および検証にも応用されている。

特にプログラミング言語への応用は盛んに研究され、これまでに多くの論理型言語が提案されている: LO (Andreoli 他, 1990), Lolli (Hodas 他, 1991), Lygon (Harland 他, 1991), ACL (小林・米澤, 1993), Forum (Miller, 1994), LLP (田村・金田, 1996), Linear LF (Cervesato 他, 1996)。これらの研究は、従来の論理型言語を表現力と効率の両面から拡張することを目指している。

しかしながら、線形論理型言語の実装方式に関する研究は進んでおらず、線形論理の応用システムを開発するための実用的なツールはほとんど見当たらない。したがって、線形論理型言語の効率の良い処理系を開発することは重要な研究課題である。

本研究では、線形論理型言語および時相線形論理型言語の効率の良い実装方式を提案し、それに基づく処理系を開発した。

- 線形論理型言語のコンパイラ処理系：
線形論理型言語のコンパイラ処理系を開発する上で重要であり、これまで解決されていなかったリソースのコンパイルについて、新しいコンパイル方式を提案した。また、その方式に基づく抽象機械および命令セットを設計した。線形論理型言語で記述した一階古典論理の自動定理証明系をベンチマークとして、開発したコンパイラ処理系の性能評価を行った結果、高速 Prolog コンパイラによる同等の定理証明系と比較して、40%の高速化を実現した。
- 線形論理型言語から Java へのトランスレータ処理系：
これまでに実現されていなかった、線形論理型言語から Java へのトランスレータ処理系を開発した。古典的な Prolog ベンチマークを用いて開発した処理系の性能評価を行った結果、既存の Prolog から Java へのトランスレータ処理系 jProlog と比較して、1.7 倍の高速化を実現した。
- 時相線形論理型言語のコンパイラ処理系：
時相線形論理に基づく論理型言語を新しく設計し、その計算モデル、コンパイラ処理系のための抽象機械および命令セットを提案した。時相線形論理型言語は、線形論理型言語の資源性と

時相論理型言語の時相性の両方を兼ねそろえており、“時刻をもったリソース”を表現することができる。

次に、本論文の構成にしたがって各章の内容について述べる。第 1 章では本研究の背景・動機・得られた成果についてまとめた。

第 2 章では線形論理の概要、線形論理型言語の設計方法、計算モデルについて述べた。線形論理型言語のほとんどは証明論的手法、すなわち“Gentzen スタイルのシーケント計算における証明探索”を用いて設計されている。特に、Miller らが提案した *Uniform Proofs* はシンプルかつ強力な設計手法であり、その代表的な設計例としては λ Prolog, Lolli, Forum がある。また、証明探索の際に生じるさまざまな非決定性についても議論し、これまでに提案されている計算モデルについて述べた。

第 3 章では線形論理プログラミングの特徴であるリソースを、線形論理型言語 Lolli を用いて説明した。

第 4 章では線形論理型言語を実装する上で重要なリソースのコンパイル方式を提案した。線形論理型言語では、リソース(使用回数を制限されたプログラム節)を実行時に動的に追加・削除することができる。しかしながら、これまでに開発された処理系では、リソースはコンパイルされず、項としてメモリ上に保持されていた。この問題を解決するために、クロージャを導入し、リソースのコンパイル方式を提案した。

第 5 章では、第 4 章で述べたリソースのコンパイル方式に基づいた、線形論理型言語のコンパイラ処理系のための抽象機械および命令セットを提案した。この抽象機械は Prolog の抽象機械 WAM の拡張になっており、効率の良いリソース管理のために新しく 24 個の命令が追加されている。

第 6 章では、線形論理型言語から Java へのトランスレート方式を提案し、その方式に基づくトランスレータ処理系を開発した。開発した処理系は、移植性、拡張性が高く、Java との双方向呼び出しが可能である。

第 7 章では、直観主義時相線形論理に基づく論理型言語を設計し、その計算モデル、コンパイラ処理系のための抽象機械および命令セットを提案した。時相線形論理では、線形論理の演算子に加え、次の時刻に一回だけ使用可能なリソースを表す様相演算子 ' \circ '、現在以降の任意の時刻に一回だけ使用可能なリソースを表す様相演算子 ' \circ '、現在以降の任意の時刻に任意の回数使用可能なリソースを表す様相演算子 ' $!$ ' を含んでいる。このため時相線形論理型言語では、線形論理型言語のリソース概念に加え、どの時刻でリソースを使用するかといったリソースの時相性を表現できる。

第 8 章では、本論文の結論と今後の課題について述べた。

| | | | |
|--|--|-----|--------|
| 氏名 | 番原 陸則 | | |
| 論文 題目 | Design and Implementation of Linear Logic Programming Languages (線形論理型言語の設計と実装) | | |
| 審査委員 | 区分 | 職名 | 氏名 |
| | 主査 | 教授 | 角田 譲 |
| | 副査 | 教授 | 金田 悠紀夫 |
| | 副査 | 教授 | 新井 敏康 |
| | 副査 | 助教授 | 田村 直之 |
| 要 旨 | | | |
| <p>本論文では、線形論理および時相線形論理に基づいた論理型プログラミング言語について、効率的な実行を目的として、新しい言語および処理系の実装方式を提案している。特に、以下の点に新規性がある。</p> <p>(1) 線形論理型言語のコンパイラ処理系: コンパイラ処理系のための抽象機械およびその命令セットの提案。</p> <p>(2) 線形論理型言語の Java へのトランスレータ処理系: Java へのトランスレート方式の提案。</p> <p>(3) 時相線形論理型言語のコンパイラ処理系: 時相線形論理型言語の設計、計算モデルの提案、コンパイラ処理系のための抽象機械およびその命令セットの提案。</p> <p>線形論理は、1987年に J.-Y. Girard が発表した新しい論理体系であり、計算機科学のさまざまな分野への応用が期待されている。線形論理の論理型言語への応用は、特に活発な分野の一つであり、Lolli, LLP などの直観主義線形論理に基づいた言語や、LO, Lygon, Forum などの古典線形論理に基づいた言語が提案されている。</p> <p>しかしながら、線形論理型言語の実装方式に関する研究は、ほとんど進んでおらず、線形論理の実際的な応用システムを研究開発する上での問題点となっていた。</p> <p>本論文の研究は、線形論理型言語の効率的な実装方式を明らかにし、実際的な応用システムの研究開発に大きく貢献したという点で高く評価できる。特に、以下の点に特徴がある。</p> <p>(1) 線形論理型言語のコンパイラ処理系 LLP: 線形論理型言語のコンパイラ処理系で最大の問題であり、これまで解決されていなかったリソースのコンパイルについて、新しい方式を提案し、その方式に基づいた抽象機械と命令セットを設計している。また、提案した抽象機械およびコンパイル方式を用いたコンパイラ処理系を実際に開発し、その性能評価を行い、これまでの処理系よりも大きく性能が改善されることを示している。</p> <p>論文中では第4章、第5章が対応する内容である。第4章で、これまで解決されていなかったリソースのコンパイルについて、関数型言語などの実装で利用されていたクロージャを導入し、リソースをコンパイルする方法を提案している。</p> | | | |

| | |
|--|-------|
| 氏名 | 番原 陸則 |
| <p>第5章では、第4章で提案したコンパイル方式に基づき、線形論理型言語コンパイラ処理系のための抽象機械 LLPAM およびその命令セットを設計、提案している。また、実際に開発したコンパイラ処理系について、複数のベンチマークプログラムをもとに性能評価した結果が示されている。特に、一階述語論理の定理証明系への応用例では、同様の既存の定理証明系より性能が大きく改善されており、線形論理型言語のコンパイラ処理系の有用性を明らかにしている。</p> <p>(2) 線形論理型言語の Java へのトランスレータ処理系 Prolog Café: これまで実現されていなかった、線形論理型言語から Java へのトランスレータ処理系を設計し、実現している。</p> <p>論文中では第6章が対応する内容である。線形論理型言語から Java へのトランスレート方式を提案し、既存の Prolog から Java へのトランスレート方式との違いおよび拡張点について述べている。また、実際に開発したトランスレータ処理系 Prolog Café について、複数のベンチマークプログラムで性能評価した結果をもとに、提案したトランスレート方式の有効性を明らかにしている。</p> <p>(3) 時相線形論理型言語のコンパイラ処理系 TLLP: 時相線形論理に基づく新しい論理型言語を設計し、その計算モデル、およびコンパイル方式を提案している。この時相線形論理言語は、線形論理の資源性と時相論理の時相性を統合した点が大きな特徴となっており、線形論理型言語でも時相論理型言語でも自然に表現できなかった「時刻を持ったリソース」を表現可能である。また、提案したコンパイル方式に基づいた抽象機械と命令セットを設計し、実際にコンパイラ処理系を開発している。</p> <p>論文中では第7章が対応する内容である。Miller の Uniform Proof および Hodas の IO モデルの概念を用い、時相線形論理に基づいた論理型言語 TLLP とその計算モデルを提案している。時相線形論理型言語 TLLP は、論理型言語 Prolog および線形論理型言語 Lolli の特徴に加え、時刻に依存したリソースを表現可能である。すなわち、様相演算子 \circ により次の時刻に一度だけ利用可能なリソースを、様相演算子 \square により現在以降の任意の時刻に一度だけ利用可能なリソースを、様相演算子 $!$ により任意の時刻に何度でも利用可能なリソースを表現することができる。また、時相線形論理型言語 TLLP のコンパイラ処理系のための抽象機械およびその命令セットを設計、提案している。さらに、実際に開発したコンパイラ処理系について、複数のベンチマークプログラムで性能評価した結果をもとに、他の実装方式よりも優れた性能をもつことを明らかにしている。</p> <p>これらの線形論理型言語処理系 LLP, Prolog Café, TLLP は、すべてインターネット上で公開されており、国内外のさまざまな研究に利用されているという点でも高く評価できる。</p> <p>以上のように、本研究は、線形論理および時相線形論理に基づいた論理型プログラミング言語について、新しい言語の設計方法と処理系の実装方式を研究したものであり、線形論理の計算機科学への応用について重要な知見を得たものとして価値ある集積であると認める。</p> <p>よって、学位申請者 番原 陸則 は、博士(工学)の学位を得る資格があると認める。</p> | |