

Title	A design and theory of strongly typed object-oriented programming languages for extensible recursive classes( Abstract_要旨 )
Author(s)	Saito, Chieri
Citation	Kyoto University (京都大学)
Issue Date	2010-03-23
URL	<a href="http://hdl.handle.net/2433/120371">http://hdl.handle.net/2433/120371</a>
Right	
Type	Thesis or Dissertation
Textversion	none

( 続紙 1 )

京都大学	博士 ( 情報 学)	氏名	齊藤智恵理
論文題目	A design and theory of strongly typed object-oriented programming languages for extensible recursive classes		
(論文内容の要旨)			
<p>本論文は、型付オブジェクト指向プログラミング言語において、再帰的クラスを安全に再利用するための新たな言語機構やそれに対する型システムを提案するものであり、以下の8章からなる。</p> <p>第1章は序章であり、オブジェクト指向プログラミング言語の特徴やその中心的概念であるオブジェクト、また型との関わりについて説明した後、既存の言語では再帰的クラスの安全な再利用が困難であるという問題に触れ、本論文で解決すべき課題を列挙し、最後に本論文の構成を示している。</p> <p>第2章は、新たな言語機構を提案する上で、基礎となる Bruce によって提案された MyType とそれに関連して厳密型(exact types)を紹介している。</p> <p>第3章は、相互再帰的クラスを安全に再利用するための枠組みである軽量族多相(lightweight family polymorphism)を提案し、それを実現させるために必要な言語機構である、相対パス型やネストクラスの同時拡張機構を提案している。また、軽量族多相の実装手法の1つとして、新たな言語機構を用いて書かれたプログラムを既存の言語機構のみで書かれたものに変換する手法を提案している。さらに、軽量族多相の型安全性や実装手法の正当性は形式的手法により証明されている。</p> <p>第4章は、第3章で提案された軽量族多相を用いる解法が、Java 言語でバージョン 5.0より採用された汎用クラス機構を用いた解法より優れる点があることを論証している。この優れた点とは、オブジェクト自身の型である自己型(self type)の表現がより継承に適していることであり、このことは形式的手法により厳密に証明されている。</p> <p>第5章は、第2章で紹介した MyType が部分型と相性が悪く、その結果動的ディスパッチを活用する機会が損なわれるという問題を指摘し、それを解決するための言語機構を提案している。手順としてまず、この問題を2つの場合に類型化し、それぞれについて解決となる言語機構が提案されている。また、それらを含む言語が型安全性定理を満たすことが証明されている。</p> <p>第6章は、汎用クラスが再帰的参照を持つが、それらを具体化する型が異なる場合、再利用に適した型を与えることが出来ないという問題を指摘し、それを解決するための言語機構である自己型構成子(self type constructors)を提案している。また、第5章での提案が、自己型構成子と組み合わせて使用すると有用であることにも触れている。自己型構成子が安全であることは、形式的手法により証明されている。</p> <p>第7章は、各章で扱った内容の関連研究について、それぞれ言及している。</p> <p>第8章は結論であり、本論文を総括している。</p> <p>本文の他に付録が4章あり、それぞれ3章から6章で示された定理の証明を詳述している。</p>			

注) 論文内容の要旨と論文審査の結果の要旨は1頁を38字×36行で作成し、合わせて、3,000字を標準とすること。

論文内容の要旨を英語で記入する場合は、400～1,100 wordsで作成し  
審査結果の要旨は日本語500～2,000字程度で作成すること。

(論文審査の結果の要旨)

近年、安全で信頼性の高いソフトウェアの迅速な開発要求が高まる中、継承を用いた再利用が可能であり、かつ実行前の型検査により安全性の保証される、型付オブジェクト指向プログラミング言語が注目を浴びている。しかし、既存の言語では、型検査が保守的に過ぎるため、再帰的参照を持つクラスを安全に再利用することが難しい。本論文では、再帰的参照を持つクラスを安全に再利用するための言語機構を提案し、その安全性を形式的に証明している。得られた主な成果は、次の通りである。

(1) 相互再帰的クラスを安全に再利用するための枠組みである軽量族多相を提案し、その安全性を形式的に証明した。この軽量族多相は、先行研究での提案と異なり依存型を用いておらず、型システムが簡潔でありながらも、実用上の表現力を十分備えている点が評価できる。

(2) 軽量族多相の実装手法の1つとして、既存のJava言語への変換手法を提案し、その正当性を形式的に証明した。これは、軽量族多相で導入された新たな機能を用いたプログラムを、既存の言語機能で模倣させる手法である。この手法の優れた点は、新たな実行環境を必要とせず、既存のJVMなどの上で実行できるという点である。

(3) 軽量族多相を用いたプログラミングがJava言語の汎用クラス機構を用いたプログラミングより優れた点があることを論証した。この研究は、汎用クラスを用いると型宣言が煩雑になるという欠点に対し、軽量族多相は単にそれに簡便な構文糖衣を与えるだけでなく、自己参照の型が安全な再利用に適している点で有利であることを示すものであり、軽量族多相という提案の意義を補強する点で重要である。

(4) 自己再帰的クラスを安全に再利用するための型であるMyTypeと部分型関係を融和させるための言語機構を提案し、それらの安全性を形式的に証明した。これは、オブジェクト指向プログラミングの特徴の1つである動的ディスパッチを活用する機会を増加させるものであり、MyTypeやその拡張を含む言語において、より柔軟で、オブジェクト指向らしいプログラミングを可能とするものである。

(5) 異なる具体型間での再帰的参照を持つ汎用クラスを安全に再利用するための型である、自己型構成子を提案し、その型安全性を形式的に証明した。この提案では、冗長な記述が必要な類似の研究と異なり、クラスの型宣言が簡潔となり、よって誤りの混入も起こりにくい点が優れている。自己型構成子を用いれば、写像メソッドを持つ汎用コレクションクラスとその部分クラスをいずれも再帰的クラスとして定義できる。よってそれらのクラスを有用なライブラリとして提供することが可能になる。

本論文により、再帰的参照を含むクラスを安全に再利用出来るプログラミング言語の開発が可能となり、ソフトウェア開発の生産性が高まるものと期待できる。よって、本論文は博士(情報学)の学位論文として価値あるものと認める。

また平成22年2月12日実施した論文内容とそれに関連した試問の結果、合格と認めた。

注) 論文審査の結果の要旨の結句には、学位論文の審査についての認定を明記すること。更に、試問の結果の要旨(例えば「平成 年 月 日論文内容とそれに関連した口頭試問を行った結果合格と認めた。」)を付け加えること。

Webでの即日公開を希望しない場合は、以下に公開可能とする日付を記入すること。

要旨公開可能日： \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日以降