様式 C-19

科学研究費補助金研究成果報告書

平成２３年 ４月１日現在

機関番号：１３３０２
研究種目：若手研究（Ｂ）
研究期間：2009 ～ 2010
課題番号：２１７０００２５
研究課題名（和文）データ管理領域におけるプログラム検証の自動化に関する研究
研究課題名（英文）Automating program verification on the data management domain
研究代表者 矢竹　健朗（YADAKE　KENRO）
北陸先端科学技術大学院大学・情報科学研究科・特任助教
研究者番号：６０４５２１１６

研究成果の概要（和文）：
本研究では､データ管理領域に焦点を当て､プログラム検証を自動化する手法を提案した。データ管理システムに特有の繰り返し処理を行う命令､推論規則を導入した言語を定義し､検証条件自動生成器を実装した。推論規則をループ不変表明に依存しないものとすることにより、検証条件の自動生成を可能とした。今後、検証条件の生成効率の改善、言語の表現力の拡張に関して検討する必要がある。

研究成果の概要（英文）：
In this research, we proposed a method to automate the program verification for the systems in the data management domain. We defined a language which has loop statements and inference rules tailored for the domain and implemented a verification condition generator. We made it possible to automatically generate verification conditions by defining inference rules free from loop invariants. We further needs to improve the efficiency of the generation and expand the expressivity of the language.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2009 年度	800,000	240,000	1,040,000
2010 年度	600,000	180,000	780,000
年度			
年度			
年度			
総　計	1,400,000	420,000	1,820,000

研究分野：形式検証技術
科研費の分科・細目：情報学・ソフトウェア
キーワード：プログラム検証、定理証明、オブジェクト指向、データ管理領域

１．研究開始当初の背景

データ管理領域とは、様々な情報システムの中でも、大量のデータを管理することを目的とするものが属す領域である。例えば、銀行システム、証券取引システムなどがこれに属す。銀行システムでは大量の顧客データや口座データが管理対象となる。証券取引システムでも同様に、株式や銘柄、売買注文といった様々なデータが管理対象となる。他にも通販システムや航空券発券システムなど、数多くのシステムがこの領域に属す。問題は、このようなシステムが扱うデータは機密度、重要度が高く、非常に慎重に扱わなければならないという点である。例えば、誤って口座を２重に登録してしまい前の残高データを消去してしまったり、大量に発行された株式売買注文の一部が処理可能容量を越えて消滅

してしまったりといったことはあってはならない。このような欠陥は、金銭面、信用面において莫大な被害をもたらすこととなる。我々が安心して安全に生活できる社会インフラを構築するためには、情報システムがデータを正しく確実に扱うことを保証することが必要不可欠である。
情報システムは最終的には何らかのプログラミング言語で実装されるため、システムの正しさを保証するためには、究極的にはプログラム検証を適用する必要がある。プログラム検証とは、プログラムの動作が論理的に正しいことを定理として証明する技法であり、プログラムを構成する様々な命令文に関する公理や推論規則を含んだ論理体系の上で行われる。その基礎はホーア論理として 1960 年代後半に確立されている。プログラム検証は非常に高い検証能力を持っている反面、自動化が難しいという問題をかかえている。それは while 文という、繰り返し処理を行う命令文の推論にユーザ介入性が必要だからである。技術的には、while 文の推論規則を使用するためには、ループ不変表明と呼ばれる、一連の繰り返しの過程で常時成立している条件を与える必要があるためである。適切なループ不変表明を自動的に求めることは難しく、通常、ユーザが考えて与えなければならない。この問題が障壁となり、プログラム検証は、迅速性が求められる現在のシステム開発では滅多に適用されていないというのが現状である。
しかし、データ管理領域においては、複雑な繰り返し処理はほとんど存在しない。大部分は、データ配列に対する単純な操作を実現するものである。例えば、銀行システムにおける利息計算の際の、口座の配列に対し一つづつ順番に利率分だけ残高を加算するといった操作や、通販サイトにおいて購入代金を決定する際の、買い物カゴの中の各商品の値段を一つづつ順番に加算し総和をとるといった操作である。このような単純操作が多いのは、データ管理系のシステムは本来、複雑なアルゴリズムにより難解な問題を解くというよりはむしろ、大量のデータに対する定型処理を自動化することを目的に作られているからである。本研究ではこの点に着目して、データ管理領域のプログラム検証を自動化することを目指す。

２．研究の目的

本研究の目的は、データ管理領域のための自動的なプログラム検証手法を構築することである。データ管理システムの特徴として、データ配列に対する定型操作が多いという特徴がある。通常、データ配列操作は、while 文という繰り返し命令によって実現される。しかし、従来のプログラム検証手法では while 文の推論にユーザ介入が必要であり、検証の自動化が困難であった。そこで本研究では、データ管理領域に見られる定型操作を直接記述するための高級な命令文を導入し、それらに関するユーザ介入性のない推論規則を定義する。これによりデータ管理領域におけるプログラム検証を自動化する。最終的に、データ管理領域のためのプログラミング言語を定義し、その検証条件生成器を実装する。 ここで、検証条件とは、プログラムが仕様を満たすために必要な、プログラム内データに関する性質のことである。本研究における「検証の自動化」とは、この検証条件を自動的に生成することを指す。生成された検証条件は、ユーザが定理証明器で証明する必要があるが、検証条件の導出に比べれば単純である。

３．研究の方法

研究は次の順序で行った。

(1)データ管理領域の分析

データ管理システム、例えば銀行システム、航空券予約システムなどの事例をもとに、データ管理領域に頻出する繰り返し操作のパターンの洗い出しを行う。

(2) データ管理言語の定義

繰り返し操作を直接記述可能な高級命令を定義し、データ管理領域のためのプログラミング言語（データ管理言語と呼ぶ）を定義する。また、システムの要求仕様を記述するための表明言語も定義する。

(3) 推論規則の導入

導入した繰り返し命令に対し、推論規則を定義する。推論規則の健全性は定理証明器 HOL で証明することにより保証する。

(4) 検証システムの実装

データ管理言語の検証システムとして、検証条件生成器の実装を行う。表明付きのデータ管理言語から検証条件を自動生成する機能を HOL の証明戦略として実装する。

(5)検証実験

検証システムの有効性確認として、実際のシステムを対象に、言語記述、検証条件生成を行い、その効率を評価する。

４．研究成果

(1) データ管理領域の分析

図書館システム、ファイアウォールシステムについて、繰り返し処理の分析を行った。その結果、全称、存在、選択、適用の 4 種類が頻繁に使用されることが分かった。全称は、配列の要素がすべてある性質を満たすかどうかを調べる操作、存在は、配列の要素に少なくともある性質を満たすものが存在するかどうかを調べる操作、選択は、配列の要素の中からある条件を満たすものだけを抽出する処理、適用は、配列のすべての要素にある処理を適用するという操作である。他にもいくつかの操作が見られたが、多くのものはこれらを組み合わせて表現することができる。

(2) データ管理言語の定義

上に挙げた 4 種類の操作を導入したプログラミング言語として、データ管理言語を定義した。データ管理言語は、システムの構成を表現しやすいようにオブジェクト指向型としている。従来の If 文に加え、全称、存在、選択、適用に対応する forall、exists、select、app という命令を導入した。While 文はループ不変表明を伴うため除外した。表明言語には、副作用を持たない、app 以外の 3 つの命令を導入した。

(3) 推論規則の導入

推論規則は、ループ不変表明を含まないように定義する。App について、ループ不変表明に依存しない推論規則を定義するためには、app に制限を与える必要がある。具体的には、app は、l を配列、x を配列中のオブジェクト、m をメソッドとすると、l->app(x|x.m())のように仕様するが、メソッド m は適用されるオブジェクト以外のオブジェクトに対してメソッド呼び出しを行ってはならない。この制限により、配列全体に対するメソッド適用の推論が、オブジェクト単体に対するメソッド適用 x.m()の推論に置き換えることができるようになる。つまり、外部メソッド呼び出しを禁止することにより、配列中のオブジェクトに対するメソッド適用が互いに影響を及ぼさなくなり、それぞれ独立して推論できるようになるということである。配列に対する推論が、オブジェクト単体に対する推論に還元されるため、ループ不変表明が不要となる。その他、forall, exists, select についてもその推論規則を導入した。これらの命令の中で使用されるメソッドは副作用を与えることが禁止されているため、容易にループ不変表明に依存しない推論規則を定義することができる。4 つの繰り返し命令を実際に定理証明器 HOL 上で定義し、その推論規則を導出できることを確認した。

(4) 検証システムの実装

検証システムとして、データ管理言語から検証条件を HOL 上の命題として生成するプログラムを開発した。まず、データ管理言語、表明を HOL の表現に変換するコンパイラを実装した。HOL では、過去の研究でオブジェクト指向理論を実装しており、今回、この理論をデータ管理言語の意味論として使用した。次に、データ管理言語の各文ごとに最弱前条件計算機能を HOL 上の証明戦略として実装した。最弱前条件とは、文の後状態を成立させる最弱の前条件のことである。最後に、それぞれの最弱前条件計算機能を連結し、入力言語全体の最弱前条件計算機能を実装した。生成される検証条件は HOL の命題として得られ、ユーザが HOL と対話的に証明することができる。

(5)検証実験

実際に、簡単な例を用いて検証システムの有効性確認を行った。例として用いたのは図書館システム、ファイアウォールシステム内のいくつかのメソッドである。実験の結果、検証条件の生成自体は自動化できるものの、その効率に問題があることが分かった。具体的には、オブジェクトに対するアクセス時に、そのオブジェクトが NULL かどうかを判定するための条件文を明示的に挿入しなければならない。最弱前条件計算では、条件文１個を処理するごとに得られる前条件の長さが２倍になるため、オブジェクトに対するアクセスの個数に対し、最終的に生成される検証条件が指数的に長くなってしまう。今後はこの問題を解決するために、NULL オブジェクトを不許可にするなど、生成効率を改善するための新たな制限を導入することを検討する。また、言語の表現力についても拡張する必要がある。現時点で、副作用のある繰り返し文は app のみであり、必ずしも表現力は高くない。ループ不変表明に依存せずに推論できる範囲でさらなる繰り返し命令を導入していく必要がある。

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計１件）

Kenro Yatake, Takuya Katayama.
An executable object-oriented semantics

and its application to firewall verification.
Software and Systems Modeling, 2010.
DOI: 10.1007/s10270-010-0160-1. (査読有)

〔学会発表〕(計１件)
矢竹健朗、データ管理領域におけるプログラム検証の自動化に関する考察、第１６５回ソフトウェア工学研究会、2009.7.3、石川

〔図書〕(計０件)

〔産業財産権〕
○出願状況 (計０件)

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
国内外の別：

○取得状況 (計０件)

名称：
発明者：
権利者：
種類：
番号：
取得年月日：
国内外の別：

〔その他〕
ホームページ等

６．研究組織
(1)研究代表者
矢竹　健朗 (YADAKE KENRO)
北陸先端科学技術大学院大学・情報科学研究科・特任助教
研究者番号：60452116

(2)研究分担者
(　　)

研究者番号：

(3)連携研究者
(　　)

研究者番号：