様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ６ 月 １０ 日現在

機関番号：８２６２６
研究種目：基盤研究(C)（一般）
研究期間：2012～2015
課題番号：２４５００００５１

研究課題名（和文）組込みソフトウェアの安全な構築のためのＣ言語のモデルとその形式検証

研究課題名（英文）Model and formal verification of the C language for secure construction of embedded software

研究代表者

Ａｆｆｅｌｄｔ　Ｒｅｙｎａｌｄ（AFFELDT, Reynald）

国立研究開発法人産業技術総合研究所・情報技術研究部門・主任研究員

研究者番号：４０４１５６４１

交付決定額（研究期間全体）：（直接経費）　　3,900,000円

研究成果の概要（和文）：ソフトウェアに対して信頼性の高い保証を与える技術として、形式検証が注目されている。しかし、ソフトウェアはCやアセンブリなどの低レベル言語で書かれると、技術的な詳細が多くなるため、現状では形式検証による安全性の完全な保証はまだ困難である。本研究では組込み応用向けのプログラムの検証のため、プラットフォームによって異なる意味論を表現し、C言語のモデルとその論理を形式化した。具体的には、共通の形式定理のライブラリに基づき、アセンブリ言語とC言語、それぞれの形式検証基盤を構築し、現実的なケーススタディーを用いてその有効性を確かめ、以上の検証実験を公開した。

研究成果の概要（英文）：Formal verification is a technique to guarantee the correctness of software with a high level of confidence. Yet, when software is written in low-level programming languages such as C or assembly, there are so many details that it is difficult in practice to perform such a verification. With embedded software as an objective, we formalize a model and a logic for C that takes platform-dependent details into account. More precisely, to allow for the verification of programs mixing C and assembly, we developed in the Coq proof-assistant a formal library whose validity is assessed by substantial case-studies.

研究分野：形式検証

キーワード：形式検証　C言語　アセンブリ言語　組込みソフトウェア　定理証明支援系　Coq

１．研究開始当初の背景

(1) 近年ではソフトウェアに対して信頼性の高い保証が求められるようになっており、その保証の方法の一つとして形式検証が注目されている。例えば、IT 製品の信頼性評価の統一的枠組みの１つであるコモンクライテリア(ISO/IEC 15408)において、最も厳密な評価保証レベル EAL7 は形式検証による安全性保証を要求する。数ある形式検証の方法の中で、最も厳密な検証が可能な手法として定理証明支援系による検証がある。

(2) 低レベルソフトウェアの定理証明支援系による形式検証のプロジェクトとして L4.verified が注目されている。L4 はマイクロカーネルを保証するために Isabelle/HOL という定理証明支援系上で C 言語のモデルとその論理（ホーア論理と詳細化のフレームワーク）を開発したが、その際その中核にあるアセンブリ言語で書かれた部分は検証の対象外になった。その他の命令型言語のソフトウェアの定理証明支援系による検証の研究もあるが、いずれも C 言語より単純化・理想化された言語を対象にしている。いずれにせよ、これらの研究は統一されたフレームワークの中で C 言語とアセンブリ言語を同時に扱おうとしていない。

(3) 分離論理は、命令型言語で書かれている低レベルソフトウェアの検証のために、開発された。分離論理は通常のホーア論理と異なりポインタの概念を自然に表すことができるので、既存研究と比べより自然な検証手法を適用できるはずである。研究開始当初まで研究代表者は分離論理のライブラリを構築し、応用実験を行ってきた。例えば、スマートカードのアセンブリ言語のモデルとその論理を提案し、応用として数理関数のライブラリの検証を行った。また、セキュリティプロトコルのネットワークパケット処理の実装の検証ができるように、C 言語のサブセットのモデルとその論理を提案した。

２．研究の目的

(1) 研究の目的は組込み用プログラムのための定理証明支援系のライブラリの構築である。研究の完成によって得られる組込み用プログラムの形式検証ライブラリは包括的なフレームワークで同時に C 言語とアセンブリ言語を扱えるので、関連研究（例えば、L4.verified）と比べ、アセンブリ言語部分についての正しさの仮定をせずに検証でき、検証結果の信頼性が高まる。

(2)関数呼出とプラットフォーム（マシンアーキテクチャ）に依存する部分を両方扱うよう、新たに C 言語のための分離論理を構築する。関数呼出のプラットフォームに依存する部分を扱えるようになると、安全性が検証されたプログラムは安全に移植できるという証拠を与えることになるので、組込み用プログラムの移植によるバグを防げることになる。

３．研究の方法

(1) 本研究では証明系の基盤として INRIA の定理証明支援系 Coq と、INRIA・マイクロソフト社の共同で開発されている Mathematical Components という拡張を利用し、その上でプログラム検証基盤をライブラリ化する。

(2) ライブラリとして分離論理の形式化を行う。C 言語においてポインタ操作は安全性上も検証においても極めて重要であるので、これを扱うための分離論理の拡充に努める。具体的には、C 言語の意味論の中にあるプラットフォームに依存する部分を抽象化して取り扱えるようにする。そうすることで、移植性のある C 言語のモデルを確保できる。

(3) 上記のライブラリでプラットフォームに依存する部分を表現したので、アセンブリ言語と同時に検証対象とすることが可能になる。アセンブリ言語の形式検証基盤を発展し、C 言語の形式検証基盤と統合させる。

(4) 上記の C 言語検証基盤とアセンブリ言語検証基盤の共通ライブラリに基づいて、C 言語とアセンブリ言語の形式検証基盤を統合すべく、関数呼出しの形式化の拡張を行う。既存のホーア論理の枠組みでは関数を扱うことは困難であるため、事前に十分な調査・検討を行う必要がある。

(5) 新たな応用で研究成果の有効性を確かめる。C 言語とアセンブリ言語で書かれている多倍長整数演算の関数実装の検証を目標とし、実用的な応用先として広く使われている GMP のコア部分を対象とする。

４．研究成果

(1)アセンブリ言語のプログラムのための形式検証基盤を完成した。その上で、詳細化という技術を用いて、この基盤上での形式検証のケーススタディーを行い、実用性を確認した [雑誌論文(1)]。ここで用いた詳細化技術は、疑似コードとアセンブリ言語プログラムの関係を形式的に表現し、それらの対応関係を用いて形式検証と容易にする技術である。具体的な成果としては、MIPS アセンブリによる符号付き多倍長整数演算関数の実装 25 個(313 行)に分離論理を適用して検証し、これらを用いた２進拡張互除法のアセンブリ言語プログラムの詳細化による検証に成功した(図 1 アセンブリ言語で検証した多倍長

整数演算関数)。

Description	
Arithmetic computations	
x←0	x unsigned
	x signed
x←1	x unsigned
	x signed
x←x / 2	x unsigned
	x signed
x←x * 2	x unsigned
x←y	x signed, y unsigned
	x, y signed
x←-x	x signed
x++	x unsigned
x←x + y	x, y unsigned
	x signed, y unsigned
z←x + y	z, x, y unsigned
	z, x signed, y unsigned
x←x - y	x, y unsigned
	x signed, y unsigned
	x, y signed
z←x - y	z, x, y unsigned
	z, x signed, y unsigned
	z, x, y signed
Arithmetic tests	
$x \overset{?}{=} y$, $x \overset{?}{<} y$, $x \overset{?}{>} y$	x, y unsigned
sign of x	x signed
parity	x unsigned
	x signed
	x, y unsigned
$x \overset{?}{=} 0$	x unsigned

図 1 アセンブリ言語で検証した多倍長整数演算関数

(2)C 言語のプログラムの形式検証基盤を完成した[雑誌論文(5)]。アセンブリ言語と違い C 言語には型や表現の豊かさの特徴があるため、扱う C 言語のサブセットを拡張した。特に、C 言語で検証を複雑にする alignment, padding の詳細を扱えるようにしたほか、しばしば安全性の問題の原因になるキャストに関して、明示的キャストのサポートを導入した。ケーススタディーとして、セキュリティプロトコル TLS の実用的実装の一部の形式検証を完成した。具体的には、定理証明支援系 Coq を用いて、ネットワークパケットのパージングを行う 161 行の関数（コメントとデバッグ情報を除き 85 行）について形式仕様を記述（形式モデル：132 行（元のプログラムのバグの修正のための 12 行を含む））し、形式検証した。最終的に、全体で約 3293 行（形式モデルの 1 行当たり約 24 行）の形式証明を記述した（図 2 ネットワークパケットのパージングの最終ゴール)。

```
Lemma POLAR_parse_client_hello_triple (SI : seq (int 8)) ...
  PolarSSLAssumptions SI →
  { init_ssl_var * init_bu * init_rb * init_id * init_ses *
    init_ciphers * init_ssl_context }
  ssl_parse_client_hello
  { error ∨ (success *
    !!((RecordHandshakeClientHello_decode SI).1)) *
    final_bu * final_rb * final_id * final_ses * init_ciphers *
    final_ssl_context }.
```

図 2 ネットワークパケットのパージングの最終ゴール

形式証明が膨大になったため、完成のために、形式検証基盤に自動検証機能を追加した。自動化に関して、国内ワークショップで発表した[学会発表(6)]し、さらに成果普及のために C 言語の形式検証基盤のデモも行った[学会発表(7)]。

また、上記の検証実験の際に形式証明が膨大になった際に、深刻な効率の問題が発生した。C 言語の形式検証基盤を拡張した際に、ツールとして用いている定理証明支援系 Coq の形式検証検査の効率（処理速度）が、予定外に大きく低下していた。平成２６年度に Coq の開発者会議に参加し議論を行い、この問題を部分的に解決できた。さらに効率の問題に対して、自動検証のためのプラグインの開発を検討し、その経験について国内ワークショップで報告した[学会発表(13)]。

(3)アセンブリ言語と C 言語の形式検証基盤の統一したフレームワークを構築した。

アセンブリ言語と C 言語が混在するプログラムの形式検証のためには、それぞれの基盤が十分な表現の詳細さを備えることが重要である。特に、アセンブリ言語と C 言語がデータ構造を共通で扱えるよう、C 言語の基本型（ポインタ、整数、構造体）の厳密な形式モデルの構築を行った。両方の検証基盤を学会で議論し[学会発表(4),(5)]、その紹介のためにフランスの IRCICA 研究所の 2XS 研究チーム（組込みソフトウェアのための安全性の研究）で成果発表を行った[学会発表(10)]。それをきっかけに、本研究の C 言語の形式検証基盤がプラットフォーム依存部分を正しく表現していて、組込みソフトウェアの形式検証に相応しいことを確認できた。

形式検証基盤を関数呼出しに対応するよう拡張した。定理証明支援系 Isabelle/HOL で行われたホーア論理の形式化[N. Schirmer. Verification of Sequential Imperative Programs in Isabelle/HOL. PhD. 2006] に基づいて、関数呼出しを明示的な意味論として直接扱うよう改良した。具体的には、再帰関数を制限なく扱えるよう、関数呼び出しを定式化し、ローカル変数の概念をモデルに明示的に導入し、健全性と相対完全性を形式的に証明した。その拡張に基づいて国際会議でチュートリアルを行った[学会発表(11)(招待講演)]。以上の呼出しの形式化に合わせて、更に、グローバル変数についても拡張し、低レベル言語に欠かせない分離論理のフレームルールを形式化した。以上の拡張によって、今後呼出し規約の概念について更に詳細化・拡張ができるようになった。

(4)本研究の多倍長整数演算関数の形式検証技術を改良した。スタンフォード大学で行われた HACS 2016 ワークショップで、最も効率的な多倍長整数演算関数実装 GMP についての形式検証プロジェクトをスペインの IMDEA Software 研究所の Dr. Strub とともに提案し

た。現在、同研究所の Dr. Dupressoir の協力を得て、形式検証実験を行っている。現時点で、多倍長整数の加算・減算・乗算・比較・シフトなどの関数実装の正当性の形式検証を完了した。これを用いてアセンブリ言語の形式検証基盤のマシン整数のライブラリを改良し、Mathematical Components に基づく新たなライブラリの開発を開始した。現在、GMP の割り算の形式検証の課題を取り組んでいる。今後検証済みの C 言語のコードの出力を行う予定である。

(5)本研究の成果普及となる講演などの活動を多く行った。定理証明支援系 Coq の入門の目的としたチュートリアル・招待講演の際、本研究を紹介し [学会発表(8),(9)]、解説記事も執筆した[雑誌論文(4)]。また、国内・外で集中講義を行った（名古屋大学大学院多元数理科学研究科理学部数理学科、2014/12/15-19；京都大学大学院理学研究科数学・数理解析専攻数理解析系、2015/7/21-24；リール 1 大学、フランス、2016/02/22-2016/03/04)。

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 6 件）

(1) Reynald Affeldt, Manabu Hagiwara, Formalization of Shannon's Theorems in SSReflect-Coq, Lecture Notes in Computer Science, 査読有, 7406 巻, 2012, 233-249
DOI: 10.1007/978-3-642-32347-8_16

(2) Reynald Affeldt. About the Formal Verification of Shannon's Theorems, Math-for-Industry Lecture, 44 巻, 2013, 86-94
https://staff.aist.go.jp/reynald.affeldt/documents/source_coding_theorem.pdf

(3) Reynald Affeldt. On Construction of a Library of Formally Verified Low-level Arithmetic Functions, Innovations in Systems and Software Engineering, 査読有, 9(2)巻, 2013, 59-77
DOI: 10.1007/s11334-013-0195-x

(4) アフェルト　レナルド、定理証明支援系に基づく形式検証、情報処理、55 巻、2014、482-491

(5) Reynald Affeldt, Kazuhiko Sakaguchi, An Intrinsic Encoding of a Subset of C and its Applications to TLS Network Packet Processing, Journal of Formalized Reasoning, 査読, 7(1)巻, 2014, 63-104
DOI: 10.6092/issn.1972-5787/4317

(6) Reynald Affeldt, An Intrinsic Encoding of a Subset of C and its Application to TLS Network Packet Processing, 日本応用数理学会 2015 年度年会予稿集（総合版）, 1 巻, 2015, 306-307

〔学会発表〕（計 13 件）

(1) Reynald Affeldt, Manabu Hagiwara, Formalization of Shannon's Theorems in SSReflect-Coq, 3rd Conference on Interactive Theorem Proving, 2012/08/14, Princeton, NJ, USA

(2) Reynald Affeldt. Formalization of Shannon's theorems, 第 8 回定理証明及び定理証明支援系ミーティング, 2012/11/22, 千葉大学

(3) Reynald Affeldt. About the Formal Verification of Shannon's Theorems (招待講演), From Modern Coding Theory to Postmodern Coding Theory, 2013/03/07, 九州大学

(4) Reynald Affeldt, Kazuhiko Sakaguchi, First Building Blocks for Implementation of Security Protocols Verified in Coq, 5th Coq Workshop, 2013/07/22, INRIA Rennes-Bretagne-Atlantique, Rennes, France

(5) アフェルト　レナルド、坂口　和彦、Coq によるセキュリティプロトコルの実装の検証、日本ソフトウェア科学会第 30 回大会、2013/09/12、東京大学（本郷キャンパス）

(6) アフェルト　レナルド、坂口　和彦、C 言語プログラムの形式検証のための Coq ライブラリの紹介、第 9 回定理証明及び定理証明支援系ミーティング、2013/11/12、信州大学工学部（若里キャンパス）、長野市

(7) アフェルト　レナルド、坂口　和彦、Coq による C プログラムの検証基盤のデモ、第 16 回プログラミングおよびプログラミング言語ワークショップ、2014/03/05、熊本県阿蘇市

(8) アフェルト　レナルド、チュートリアル：定理証明支援系 Coq 入門（招待講演)、日本ソフトウェア科学会 31 回大会、2014/09/07、名古屋大学東山キャンパス

(9) アフェルト　レナルド、定理証明支援系 Coq による形式検証（招待講演)、情報処理学会第 77 回全国大会、2015/03/17、京都大学吉田キャンパス

(10) Reynald Affeldt, First Building Blocks for Implementations of Security Protocols Verified in Coq (招待講演), CRISTAL Seminar, 2015/07/10, IRCICA 研究所 (Villeneuve d'Ascq, France)

(11) Reynald Affeldt, Proving Properties on Programs (招待講演), Coq Tutorial at ITP'15, 2015/08/29, Hanyuan Hotel (Nanjing, China)

(12) Reynald Affeldt, An Intrinsic Encoding of a Subset of C and its Application to TLS Network Packet Processing (招待講演), 日本応用数理学会 2015 年度年会, 研究部会 OS: 数理的技法による情報セキュリティ, 2015/09/11, 金沢大学 (石川県金沢市)

(13) Reynald Affeldt, Coq Coding Sprint 参加報告, 第 11 回定理証明及び定理証明支援系ミーティング, 2015/09/17, 神奈川大学 (神奈川県横浜市)

〔その他〕
ホームページ:
https://staff.aist.go.jp/reynald.affeldt/seplog/

６．研究組織
(1)研究代表者
アフェルト　レナルド (AFFELDT Reynald)
産業技術総合研究所・情報技術研究部門・主任研究員
研究者番号：40415641

(2)研究分担者
大岩　寛　　(OIWA Yutaka)
産業技術総合研究所・情報技術研究部門・研究グループ長
研究者番号： 20415649

(3)連携研究者　なし