様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成２５年　５月３１日現在

機関番号：１２６０８
研究種目：基盤研究（Ｂ）
研究期間：2010～2012
課題番号：22300007
研究課題名（和文）ＣＯＩＮＳコンパイラの最適化器の時相論理による生成・検証と最適化器の高度化拡張
研究課題名（英文）Generation and verification of COINS compiler optimizers using temporal logic and high-level extensions of optimizers
研究代表者
佐々　政孝（SASSA　MASATAKA）
東京工業大学・大学院情報理工学研究科・教授
研究者番号：20016182

研究成果の概要（和文）：
ＣＯＩＮＳコンパイラの低水準中間表現に対し，双方向時相論理であるCTL-FVを用いて最適化のパターンと変換を記述し，そこからモデル検査により，Ｃ言語の最適化器を生成した．
一方，通常形式上の部分冗長除去（PRE）を静的単一代入形式上で行うように変換する汎用的な手法を開発した．また，PREおよびPREに似た部分無用コード除去法の効率化と効果の向上を行い，PREの応用として，同じ配列を参照するロード命令のキャッシュヒット率を向上させる手法を実現した．

研究成果の概要（英文）：
We generated a C language compiler optimizers using the low-level intermediate form of the COINS compiler. We specify patterns and transformations of optimizers in CTL-FV, which is a bi-directional temporal logic, and performed model checking.
On the other hand, a generic algorithm is made, which converts partial redundancy elimination (PRE) in the normal form into one in the static single assignment form. In addition to the generalization, we have improved the effectiveness and the efficiency of the PRE and the partial dead code elimination which is another code optimization technique similar to PRE. As a new code optimization, we also have proposed a technique that increases cache-hit ratio.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2010 年度	5,900,000	1,770,000	7,670,000
2011 年度	4,600,000	1,380,000	5,980,000
2012 年度	3,400,000	1,020,000	4,420,000
年度			0
年度			0
総　計	13,900,000	4,170,000	18,070,000

研究分野：総合領域
科研費の分科・細目：情報学・ソフトウエア
キーワード：コンパイラ，コード最適化，時相論理，静的単一代入形式，網羅型データフロー解析，要求駆動型データフロー解析，キャッシュ効率化，大域値番号付け

１．研究開始当初の背景

プログラムの実行速度向上のためには，コンパイラの最適化が不可欠である．我々は多くの研究者，開発者が利用できる，コンパイラ・インフラストラクチャ COINS [http://sourceforge.net/projects/coins-project/]を作成し，その最適化器［佐々ほか:情報処理学会論文誌:プログラミング,2006］の研究，開発を継続し，その充実を計ってきた．しかし，次のような課題が残された．

(1)最適化器を正しく実装する作業には多くの苦労があり，結果として，解決されていないバグもある．

(2) さらなる最適化のアイディアはあったがまだ実現されていないものも多い．

(1) についての研究は，次の３つに分類できる．①最適化器そのものが正しいことを検証する．検証された最適化器は，任意のプログラムについて，その振舞いを変えることなく最適化できる．②テストプログラムを用いて，最適化器がそのテストで引き起こした中間表現の変化などを検証する．③テストプログラムを用いて，最適化によって変化したテストコードの実行を確かめる．

①としては,Lacey らの研究やLernerらの研究などがある．これらの研究は証明できるドメイン言語を定義することによって，理論上の厳密性を保証するが，実際にあまりバグが混入しにくい簡単な最適化しか扱えず，実用性に欠ける点がある．

②としては，Necula や佐原らの研究がある．Necula の研究は最適化前後のプログラムにおいて，記号実行をしながら評価を行い,各対応点(call 文,jump 文および return 文）まで結果がすべて等価であれば，プログラムの変換も正しいとする．この検査手法は，高速で現実的であり，どんな最適化にも使える．しかし，判定不能の場合，推測を用いるため，厳密なプログラムの意味の保存を保証できない．この手法は複雑な最適化に対しては精度が低いと思われる．

③としては，Jaramillo らの研究などがある．この研究は最適化前後のプログラムを交互に実行して行き，対応する変数に対する値が一致しているかどうかを検証する手法である．しかし，理論上の裏付けが弱い．たとえば，ある実行に対して，最適化前後の変数の値が一致したとしても，すべての実行に対して，この変数の値が一致することが保証できない．逆に，乱数を使うプログラムのような場合は，正しい変換による最適化前後の変数の値は異なるかもしれないため，バグでないものをバグとして検出してしまう．また，トレースデータが大きすぎ，検証時間が長い，などの欠点がある．

本研究は，これらのいずれにも属せず，最適化器の仕様を時相論理によって記述し，そこからコンパイラの最適化器を生成するものである．類似の研究として，Lacey らの研究があるが，彼らはごく小さな言語について実装しただけである．本研究では，C 言語を対象として実装を行う点に特長と実用性がある．

(2)の分類にあたるものとしては，申請者らは，静的単一代入形式(以下，SSA 形式という)上の最適化を研究してきた．SSA 形式は，最適化が誤りなく容易に行える形式として注目されているが，国内外でもまだ十分な研究が行われているとは言えない．また，通常形式の低水準言語において適用できる最適化についてもいくつかのアイディアがあるが，まだ実現されていない．

申請者は，次のような研究を行ってきた．①質問伝播に基づく投機的部分冗長除去［情報処理学会論文誌プログラミング，2009］，②自動的等価性差分の抽出による SSA コンパイラ最適化器の生成するコードの正しさの検証［情報処理学会論文誌プログラミング，2009］,③要求駆動型部分無用コード除去［組込みシステムシンポジウム，2008］， ④COINS コンパイラ・インフラストラクチャの開発［コンピュータソフトウェア,2008］. ⑤Generating Java Compiler Optimizers Using Bidirectional CTL［Electronic Notes in Theoretical Computer Science，2007］.

本研究は，これらの経験の上に，時相論理

によるＣ言語コンパイラの最適化器の生成，および，静的単一代入形式あるいは通常形式の低水準中間表現上での新しい最適化の実現を目指すものである．

２．研究の目的

「１．研究開始当初の背景」で述べた課題を解決するため，本研究では次を行う．

(1) コンパイラにおける時相論理を用いた　最適化器の生成の研究

(2) SSA 形式最適化の新しい方式，通常形式最適化の新しい方式の研究

３．研究の方法

基本的な流れは次のとおりである．

(1)サーベイを行い，従来法の問題点を洗い出す

(2)プロトタイプのアルゴリズムの設計

(3)プロトタイプの実装

(4)プロトタイプの評価

(5)学会大会等で発表，意見をもらう

(6)本システムの設計

(7)本システムの実装

(8)本システムの評価

(9)査読付き論文誌等に発表

必要に応じて後戻りを行う．

なお，モデル検査器は自作する．

新しい最適化は，コンパイラ・インフラストラクチャ COINS の上に作成する．

４．研究成果

(1) 双方向 CTL による C 言語コンパイラの最適化器の生成

近年，時相論理によるコンパイラ最適化器生成の研究が行われている．たとえば，Fang や Warburton らの研究である．しかし，これらは Java 言語を対象としており，Ｃ言語を対象として最適化時間や目的コードの実行時間を提示した研究はない．

本研究では，従来研究で欠けていた点を克服し，時相論理による記述から，コンパイラ・インフラストラクチャであるCOINS上で，Ｃ言語の効率良い最適化器を生成するシステムのプロトタイプを作成した．用いる論理としては，分岐時相論理の1つで過去時制と未来時制をともに扱える双方向CTLであるCTL-FVを用いた．

最適化変換を記す仕様記述も，従来法と異なり，条件式を満たす特定の番号の個々の命令文ではなく，命令文の集合を計算するようにしたので，最適化が容易に記述できるようになった．さらに，本研究は一時変数を記述できるようにするなど種々の実用化の工夫を行った．

本研究が実装したモデル検査器は何の変換も行わずに，過去時制と未来時制を直接扱える．

なお，既存研究では，システムの実装において扱う中間言語が高級言語に近い言語であるのに対して，本研究では，より機械語に近い中間言語であるLIR というものを対象としたシステムを実装した．

作成したコンパイラ最適化器は，

・無用命令除去
・コピー伝播（定数伝播を含む）

である．

この最適化器の生成は次の部分からなる．

・コードのモデル化
　低水準中間コードの制御フローグラフを用いる
・最適化検査仕様の作成
　これは次のような形式で作成する．

MATCH
変数= 式
CONDITION
point 文字列: CTL-FV 式
edge 文字列: *point* 文字列→ *point* 文字列
PROCESS
point 文字列: *Delete* 命令文
point 文字列: *Replace* 式→ 式

これは，MATCH式にマッチし，CONDITIONを満たす中間表現の部分に対し，PROCESSで指定した変換を行うものである．

・自由変数の束縛
　上記の最適化検査仕様にあらわれる変数は自由変数である．これを実際の変数に束縛する．
・モデル検査と書き換え作業
　MATCH部に適合する命令を書き換える．
・実験
　SPEC CPU 2000から７つのベンチマークを選び，実験を行った．
　コピー伝播と無用命令除去の最適化器を生成した．これらの最適化後の目的コードの実行時間は，もともとのCOINSに備わってい

た同種の手書きの最適化とほぼ同じであったので，手書きの最適化器と同等の効果があると言える．
しかし，最適化時間は，手書きのものに比べて遅いという結果が得られた．

・工夫した点
生成される最適化器の処理時間を短くするための工夫や，記述のノウハウなど，本手法の改善に向けた種々の考察を加えた．

・今後の課題
時相論理から生成した最適化器はコピー伝播と無用命令除去である．その他の最適化器，たとえば部分冗長除去の扱い，は未だ行っていない．これらへの適用，および最適化時間の低減が今後の課題である．

・まとめ
通常の手書きの最適化器と同等の性能を持った，時相論理CTL-FV によるＣ言語コンパイラの最適化器の生成は本研究が初めてである．
その可能性と今後の課題を明らかにした．

(2) 静的単一代入形式上で通常形式部分冗長除去を実現する汎用的手法

部分冗長除去(PRE)は，部分的に冗長な式を除去する変換で，共通部分式除去とループ不変式移動の効果を含んだ効果的なプログラム最適化である．この PRE を，最適化に適した形式である静的単一代入(SSA) 形式上で行おうとする従来研究がいくつかあるが，これらは難解である上，一般には容易ではない．その理由は，SSA 形式の特徴である，変数名の一意性に関する規則が PRE の実装の妨げになるからである．例えば，同じ名前であった変数同士が SSA 形式化に伴い異なる名前になり，変数名の同一性の判断ができなくなる，といった問題が挙げられる．このような問題に対し，従来手法では，PRE を SSA 形式上で実現するために特別なデータ構造を用いるなど複雑な処理を行っていた．

これに対し本研究では，SSA 形式でありながら通常形式にも近い性質を持つ CSSA 形式と phi congruence class というものを利用すれば，SSA 形式でも通常形式における変数名の同一性が判断できる事に着目した．その事実に基づいて，通常形式の PRE アルゴリズムを SSA 形式に適用する手法を提案した．この手法は汎用的なものであり，挿入点の決定・式の挿入・式の置き換え（冗長性の除去）という通常の手順に従う PRE であれば，原則としてアルゴリズムに依らず SSA 形式に対応させることが出来，また元々の PRE のアルゴリズムの枠組みを変える必要もない．

本手法を確かめる実験として，代表的な PRE アルゴリズムの一つであるLazy Code Motion を SSA 形式上のアルゴリズムに変換し，変換後でも部分冗長除去の効果が変わりなく発揮されていることを確認した．

(3) 大域値番号付けに基づく要求駆動型部分冗長除去法

PRE は，その適用によって，部分冗長な式を除去するだけでなく，字面からは分からなかった部分冗長な式を明らかにする副次的効果をもっている．この PRE の適用によって生じる副次的効果を反映させるためには，PRE の複数回の適用が必要である．しかしながら，PRE は，プログラム全体を解析する網羅型データフロー解析に基づいているので，PRE の複数回適用は，コストが高いことが問題であった．

解析コストを抑えながら副次的効果を反映する手法の一つに，滝本らが提案した要求駆動型データフロー解析を用いて解析範囲を限定した要求駆動型 PRE がある．要求駆動型 PRE を要求駆動型コピー伝播とともに，開始点に近いものから順に，各式に適用することによって，多くの副次的効果を反映させることができる．しかしながら，要求駆動型のコピー伝播は，変数の生存期間を伸長する傾向があり，レジスタ圧力を高めるので，レジスタスピルを生じさせる可能性があった．

本手法は，演算子とオペランドの変数という式の字面で冗長な式を検査する代わりに，演算子とオペランドの値番号によって式を検査するので，コピー伝播による字面の変形を行うことなく副次的効果を反映することができる．

本手法を COINS の最適化器として実現し，ベンチマークプログラムに適用して効果を調べたところ，従来の副次的効果を反映させる手法よりも，目的コードの実行効率を向上させることに成功した． 現在，本手法を用いて，ループの繰返し間で同じ値を保持する配列参照を同じレジスタで置き換えるスカラ置換の実現を進めている．本手法によって，任意の制御構造をもつループを含め，プログラム全体に対して，スカラ置換が実現できる．

(4) 要求駆動型部分無用コード除去法

部分無用コード除去（PDE）は，実行経路によっては無用であるが，その他の経路では無用でない代入文を除去する手法である．PDE は，その適用の結果として，PRE の場合とよく似た副次的効果を生じる可能性があるので，PDE の繰返し適用が効果的であるが，PRE の場合と同様に網羅型データフロー解析に基づいた解析法を用いているので，解析コストが高いという問題があった．

本手法は，PDE を要求駆動型データフロー解析によって実現し，終了点に近い代入文から順に適用することによって，解析コストを抑えながら，多くの副次的効果を反映させる．

本手法を COINS 上に実現し，ベンチマークプログラムに適用したところ，PDE を 1 回適用する解析コストで，多くの副次的効果を反映させることができることを確認した．

(5) ロード命令集約によるキャッシュヒット率向上手法

現在の多くのコンピュータは，高速な CPU と低速な主メモリによって構成されており，プログラムの実行中に主メモリにアクセスするロード命令が実行されると，プログラムの実行効率が低減される可能性がある．そこで，多くの CPU には，キャッシュメモリが備えられており，ロード命令を実行する際は，まず，キャッシュメモリにアクセスアドレスの値が保持されていないかチェックする．一旦，主メモリへのアクセスが生じると，アクセス箇所の近傍の値がキャッシュメモリにコピーされる．次にロード命令が実行される際に，キャッシュメモリにアクセスアドレスの値が見つかりキャッシュがヒットすれば，ロード命令は，主メモリにアクセスすることなく効率的に実行できる．一方，キャッシュにアクセスアドレスの値が見つからずキャッシュミスすれば，主メモリへのアクセスが生じ，実行が非効率になる．

本手法は，同じ配列にアクセスするロード命令が連続して実行されるように集約することで，キャッシュのヒット率を向上させる．このロード命令の集約は，PRE の同じ値を生成する式を解析する過程を，同じ配列にアクセスするロード命令を解析するように置き換えることで実現する．本手法は，Lazy Code Motion のアルゴリズムを採用しているので，集約が可能でなければ，プログラムの変形は行わない．また，集約が可能であった場合でも，不要な巻上げを避ける効果がある．

本手法を COINS の最適化器として実現し，ベンチマークプログラムに適用して効果を調べたところ，キャッシュのヒット率を向上できることを確認した．

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 4 件）

① 澄川靖信，滝本宗宏：効率的な要求駆動型部分冗長除去，情報処理学会論文誌：プログラミング，掲載予定，(2013 年)，査読有り

② 澄川靖信，滝本宗宏：配列の次元を考慮した大域ロード命令集約，信学技報，電子情報通信学会，Vol.112，No.164，SS2012-29，pp.115-119，(2012 年 7 月)，査読無し

③ Takimoto, M.: Demand-driven Partial Dead Code Elimination, IPSJ Transactions on Programming Vol.5, No.1, pp.9-16, (Mar. 2012), 査読有り

④ Sassa, M., Imahashi, T. and Ito, Y.: A Generalized Method for Realizing Partial Redundancy Elimination for Normal Forms in Static Single Assignment Forms, Advances in Computer Science and Engineering, Vol. 7, No. 1, pp. 1-24,(Aug. 2011).査読有り

〔国際会議プロシーディングス〕（計１件）

① Sumikawa, Y. and Takimoto, M.: Global Load Instruction Aggregation Based on Code Motion, Proc. of IEEE International Symposium on Parallel Architectures, Algorithms and Programming, PAAP'12, IEEE Computer Society, pp.149-156, (Dec. 2012). 査読有り

〔学会発表〕（計６件）

① 澄川靖信，滝本宗宏：スピルコストを考慮した部分冗長除去，情報処理学会第 75 回全国大会講演論文集，情報処理学会，Vol.2013，No.1，pp.345-346，(2013 年 3 月 8 日）仙台.

② 澄川靖信，滝本宗宏：質問伝播に基づく大域ロード命令集約，第 54 回プログラミングシンポジウム，情報処理学会，pp.19-26，(2013 年 1 月 11 日)，強羅.

③ 澄川靖信，滝本宗宏：効率的な要求駆動

型部分冗長除去，情報処理学会第 74 回全国大会講演論文集，情報処理学会，Vol.2012，No.1，pp.427-429，(2012 年 3 月 6 日)，名古屋.
④ 新屋良磨，光成滋生，佐々政孝：並列化と実行時コード生成を用いた正規表現マッチングの高速化，情報処理学会プログラミングシンポジウム，第５３回，p.159-162 (2012 年 1 月 8 日)，湯河原，ポスター発表.
⑤ 澄川靖信，滝本宗宏：コード移動に基づく大域ロード命令集約，日本ソフトウェア科学会第 28 回大会講演論文集，日本ソフトウェア科学会 (2011 年 9 月 29 日)，那覇.
⑥ 新屋良磨，光成滋生，佐々政孝：並列化と実行時コード生成を用いた正規表現マッチングの高速化，日本ソフトウェア科学会大会論文集，第２８回，6A-2 (2011 年 9 月 24 日)，那覇.

〔その他〕
ホームページ
http://www.is.titech.ac.jp/~sassa/coins-www-ssa/japanese/index.html

６．研究組織
(1)研究代表者
佐々　政孝 (SASSA MASATAKA)
東京工業大学・大学院情報理工学研究科・教授
研究者番号：２００１６１８２

(2)研究分担者
滝本　宗宏 (TAKIMOTO MUNEHIRO)
東京理科大学・理工学部・准教授
研究者番号：００３１８２０