

令和 2 年 5 月 22 日現在

機関番号：33917

研究種目：基盤研究(C)（一般）

研究期間：2017～2019

課題番号：17K00114

研究課題名（和文）部分評価を用いたWebアプリケーションの横断的依存解析手法に関する研究

研究課題名（英文）Cross-sectional dependency analysis method based on the partial evaluation for Web application

研究代表者

吉田 敦 (Yoshida, Atsushi)

南山大学・国際教養学部・教授

研究者番号：50283495

交付決定額（研究期間全体）：（直接経費） 2,900,000円

研究成果の概要（和文）：フレームワークを利用したウェブアプリケーションの開発支援として、部分評価に基づく横断的依存解析手法の開発に取り組んだ。PHPの構文解析器の構築、部分評価器の構築、横断的な依存解析の実現の順に取り組みを行なった。構文解析器はある程度実現できたが、部分評価器はプロトタイプの改善に留まり、依存解析そのものは実現できなかった。依存解析の支援として、例外処理のように解析から外す枝刈りを実現するために、例外処理を自動的に特定する手法を実現した。

研究成果の学術的意義や社会的意義

Webアプリケーションは、社会の主要なインフラを構成する要素であり、高い品質で開発することが必要である。そのためには、依存解析などの開発支援が必要であるが、ウェブアプリケーションの場合、複数の言語による記述で構成されることで構文解析が難しく、また、動的なリフレクションが適用されるなど依存解析も容易ではない。本研究は、そのための基盤を構成することで、解析支援環境の構築を容易にする。

研究成果の概要（英文）：To support the development of web applications using frameworks, we have developed a cross-sectional dependency analysis method based on the partial evaluation. We tried to implement in the following order: construction of PHP parser, construction of partial evaluator, and realization of cross-sectional dependency analysis. We have realized the PHP parser to some extent, but we have only slightly improved the prototype of the partial evaluator. We could not realize the dependency analysis itself. Instead, we have developed a method to automatically identify exception handlings to remove the branches in the syntax tree for dependency analysis.

研究分野：ソフトウェア工学

キーワード：Webアプリケーション PHP 構文解析 依存解析 部分評価

1. 研究開始当初の背景

Web アプリケーションの開発において、サーバーサイドで使われるプログラミング言語の代表として PHP がある。PHP は、ページ記述の中にコードを埋め込むことができ、動的なページを記述しやすいといった利点がある。また、リフレクションやマジックメソッドにより、実行時により柔軟な制御も可能になっている。さらに、PHP でのアプリケーション開発を容易にするために、MVC などのパターンに基づく様々なフレームワークも開発されており、それらを使うことで、ページ間の遷移の制御やデータの受け渡しが簡単に実現できるようになっている。一方、テストなどでバグがあることがわかって、入力から出力の間の制御およびデータの流れが追跡しにくいという問題が生じる。特にリフレクションやマジックメソッドを用いると、静的な解析が困難となり、支援ツールの構成が難しくなる。また、フレームワークの挙動を正確に理解していないと、目視による追跡も難しいことがある。フレームワークを初めて学ぶ学習者は、依存関係を読み取れないために、バグがあることがわかって修正方法がわからないこともある。

2. 研究の目的

本研究では、部分評価を利用し、フレームワークで用いられるコンポーネント間を横断的に依存解析するための方法と、それを用いた支援方法を明らかにしようとした。部分評価を用いるのは、動的な解析と静的な解析の中間的な解析を行うことで、リフレクションやマジックメソッドの問題を回避できると考えられるからである。

3. 研究の方法

PHP のアプリケーションを対象として解析するために、(1) 構文解析系の実現、(2) 部分評価の方法の検討とその実現、(3) 横断的解析の検討と実現を行う。

- (1) 構文解析系の実現にあたっては、すでに実現している C 言語の解析系を応用する。この解析系は属性付き字句列の書換え規則で構成され、コード片の解析ができる。また、ボトムアップ的に解析をするので、コード片以外の記述が混在しても、そのコード片を構文解析できる。Web アプリケーションのフレームワークを用いたプログラムの場合には、このような機能が有効であると考えられる。
- (2) 部分評価の方法の検討とその実現にあたっては、すでに構成済みのプロトタイプを拡張する。ただし、このプロトタイプは対応する構文要素や式表現などが限定されているので、全体の設計の見直しが必要である。
- (3) 横断的解析の検討と実現にあたっては、部分評価によって何ができるのか、どのような解析支援が有効かを検討しながら進めていく。

4. 研究成果

- (1) 構文解析系の実現にあたっては、C 言語の解析系を参考に開発を行なった。C 言語と異なり、変数識別子と型識別子を構文解析の過程で区別する必要がないので、ボトムアップな解析を素直に実装できた。ただし、当初は PHP の基本的な構文、すなわち、最近の構文の拡張を考慮しない形で進め、そのあと、最新の構文について対応していった。この報告書作成時点では約 220 行の書換えルールで構成されている。PHP は、当初は動的型付けの言語であったが、型付けのための構文要素が加わり、複雑化している。加えて、予約語 use のように、複数の文脈で異なる目的で使われる字句があり、コード片を対象とした部分的な構文解析を難しくしていた。これらのことから、古くからある構文規則には対応できているものの、最新の構文規則には追いついていない状況である。
- (2) 部分評価については、当初、プロトタイプの拡張を予定していたが、評価のルールを整理するために、あらたに構文解析の結果から評価用の構文木を構成する方法に変更した。構文解析器は、属性付き字句列の形式で構文木を構成するが、評価においては、グラフとしての木構造を実装した方が容易になる。また、複数のツール間の連携を容易にするために、一般的なデータフォーマットを使用することを検討した。その結果、JSON 形式のデータフォーマットを選択し、各節点が属性付き字句列を保持し、かつ、その中に子要素が入る形の書式を採用した。これにより、JSON 形式のデータから、字句列だけを抜き出して結合すれば元のソースコードになり、また、JSON のデータをプログラム中でデコードするのみで、メモリ上に構文木がすぐに構成できるようになった。なお、(1) の構文解析系の作業と並行に行なっていたが、(1) の作業に多くの時間が費やされ、この構文木上での部分評価系の実装には至らなかった。なお、これらの取組みは、C 言語の解析系にも適用し、プログラミングの教育支援の研究に応用された。
- (3) 横断的解析については、(2) が実現できなかったため、当初の目標は達成できなかった。一方、各作業を進める中で、アプリケーションのソースコードの理解を支援する方法として、

ソースコードの読み手にとって不要な部分を省略するという方法があるとの考えに至り、それについて取り組んだ。多くの場合、基本的な処理の流れを理解するには、例外処理は不要であり、例外処理となる条件文を特定する方法に絞って取り組んだ。例外処理は、ライブラリ関数の返り値の処理で、エラー判定をして、エラーメッセージを出力する。この例外処理の書き方は、アプリケーションごとには異なる可能性があるが、アプリケーション内では全体に統一的に書かれることから、同じような部分字句列(n-gram)をより多く持つ条件文は例外処理である可能性が高いと想定し、その調査を行なった。その結果、6~7割程度は特定できることを確認した。さらに、条件文に限定しないで、任意の文の列に対して、アプリケーション内で共通する部分字句列を含むものを探すと、初期化やオプション処理などを特定できる可能性があることも確認をした。ただし、これは複数のツールから構成されるアプリケーションの場合であり、アプリケーションごとに特定できるものは変わると想定される。また、このとき、構文木の構造に基づく n-gram を構成することで、精度を向上できる可能性があることもわかった。これらの実験は、C 言語で行ない、PHP では行なえなかったが、言語間の差異はないと想定されるので、横断的な依存解析を実現するときに、枝刈りに使えると考えられる。

5. 主な発表論文等

〔雑誌論文〕 計7件（うち査読付論文 5件 / うち国際共著 0件 / うちオープンアクセス 0件）

1. 著者名 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋	4. 巻 Vol.35, No.4
2. 論文標題 プログラミング演習用ブルーフリーダの試作	5. 発行年 2018年
3. 雑誌名 コンピュータソフトウェア	6. 最初と最後の頁 129-135
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.35.129	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 吉田敦, 加藤大貴, 蜂巢吉成, 桑原寛明, 阿草清滋	4. 巻 2018
2. 論文標題 字句列の共通性に基づく例外処理条件文の抽出手法の提案	5. 発行年 2018年
3. 雑誌名 ソフトウェアエンジニアリングシンポジウム2018論文集	6. 最初と最後の頁 15-23
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 石元慎太郎, 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋	4. 巻 2018
2. 論文標題 プログラミング演習における構文要素の種類毎のビューによるコーディング状況把握方法の提案	5. 発行年 2018年
3. 雑誌名 情報教育シンポジウム論文集	6. 最初と最後の頁 158-165
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 安達有希, 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋	4. 巻 2018
2. 論文標題 プログラミング学習における構文図式を用いた構文理解支援方法の提案	5. 発行年 2018年
3. 雑誌名 情報教育シンポジウム論文集	6. 最初と最後の頁 118-125
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 加藤宗一郎, 吉田敦, 蜂巢吉成, 桑原寛明, 阿草清滋	4. 巻 44
2. 論文標題 記述の共通性に着目したプログラムダイジェスト化手法の提案	5. 発行年 2018年
3. 雑誌名 ソフトウェア工学の基礎XVII レクチャーノート/ソフトウェア学 44	6. 最初と最後の頁 155-156
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋	4. 巻 43
2. 論文標題 プログラミング演習用ブルーフリーダの試作	5. 発行年 2017年
3. 雑誌名 ソフトウェア工学の基礎XXIV	6. 最初と最後の頁 53-62
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 蜂巢吉成, 小林 悟, 吉田敦, 阿草清滋	4. 巻 Vol. 3, No.1
2. 論文標題 プログラミング演習におけるテストケース評価システムの提案	5. 発行年 2017年
3. 雑誌名 コンピュータソフトウェア	6. 最初と最後の頁 64-78
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計1件 (うち招待講演 0件 / うち国際学会 0件)

1. 発表者名 加藤大貴, 吉田敦, 蜂巢吉成, 桑原寛明, 阿草清滋
2. 発表標題 APIに対する例外処理の共通的特徴に基づくコード例の推薦手法の提案
3. 学会等名 日本ソフトウェア科学会 FOSE2017
4. 発表年 2017年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究 分担 者	蜂巢 吉成 (Hachisu Yoshinari) (30319298)	南山大学・理工学部・教授 (33917)	