学位論文要旨

学位論文題目　C言語プログラムの類似度評価に関する研究

申請者氏名　包　胡日查

プログラムソースコードの類似性を判定することはプログラム本体の重複コードや無駄なコードを減らせ，さらに知的財産の分野では，著者の知識財産を守ることも意味している．C言語プログラムは手続き言語であり，実行順で関数の呼び出し関係ができ，関数内部でも順に実行されるので全プログラムを1つの木グラフとして表現し，木グラフのノード節点はCソースコードで表現できる．

木構造はXMLファイルや遺伝子の進化過程など，様々なオブジェクトの表現に用いられている．このような木で表現されたオブジェクトを検索したり分類したりするには，木の類似性判定が必要である．プログラミング教育の現場において，教師は学生から提出される大量のプログラムを目視により妥当性を判断し，それぞれのプログラムに適した評価をする必要がある．

二つのC言語プログラムがどれぐらい類似しているか，文字列を厳密に比較して算出するのではなく，プログラムを構文木に変換し，変換された木グラフの類似度を用いてプログラムの類似度を算出する．構文木に変換して比較することは，計算コストを下げるだけではなく，木の特性を生かした比較ができる．これにより，文字列比較ではできなかったプログラム構造の比較が可能になり，また関数の呼び出し関係まで比較することができる．我々はC言語プログラム同士の類似度を測ることを目的し，C言語プログラムの類似度の求める方法としては，New Tree Overlapping 手法とDepth Matching 手法を提案する．

本論文は以下のように第1章から第6章までで構成される．

1章では研究の背景を紹介し，関連の研究状況を概説しており，また研究の目的について述べる．

2章では，研究に関連する基礎的な定義や概念を記述する．まず，木グラフ(Tree Graph)の定義や構文木（Syntax Tree）や二部グラフのマッチング（Bipartite Matching）について説明した上で，それらの基本的な性質を示す．次に文字列編集距離と木間の編集距離について説明し，最後に既存のプログラム類似性判定手法のSMMTについて紹介する．

3章では，C言語プログラムを構文木化手法を提案する．まずはC言語ソースプログラムについて前処理を行う．次に我々はC言語プログラム構成のパターンを分析し，分析したパターンをすべて表現できる構文木部品を提案し，これらの部品を使いC言語プログラム全体を木グラフに変換するCCXソフトを提案する．最後に変換手法について例を用いて説明する．

4章では，構文木グラフの類似度を求める手法を提案する．既に提案されたTree Overlapping手法を紹介した上で，すべての共通部分木を用いて構文木の類似度を測るNew Tree Overlapping（NTO）方法を提案する．また，木グラフの深さを基準にノード間の最大マッチングを計算することによって構文木の類似度を測るDepth Matching（DM）手法を提案する．

5章では，サンプルプログラムを用いて提案した二つの手法と既存のSMMT手法との比較実験を行い，その結果について議論する．サンプルプログラムは，学生のレポートやテキストに掲載のプログラムや研究用のプログラムなどである．実験結果から，提案のDM手法が最も良く，その有効性が確認できた．

6章では，本論文で得られた結果をまとめており，今後の展望や課題について述べる．

（様式９）

学位論文審査の概要と結果

報告番号	東アジア博 甲 第 95 号	氏　名	包　胡日查
論文題目	C 言語プログラムの類似度評価に関する研究		

（論文審査概要）

本論文は，C 言語プログラムの制御構造を考慮したプログラムの類似性を評価するために，木グラフを用いて，New Tree Overlapping（NTO）と Depth Matching（DM）の方法を提案したものであり，６章の構成になっている．

１章では学位論文の研究背景を紹介し，関連の研究状況を概説しており，また研究の目的について述べている．

２章では，研究に関連する基礎的な定義や概念を記述している．まず，木グラフ(Tree Graph)の定義や構文木（Syntax Tree）や二部グラフのマッチング（Bipartite Matching）について説明した上で，それらの基本的な性質を示している．次に文字列編集距離と木間の編集距離について説明し，最後に既存のプログラム類似性判定手法の SMMT について紹介している．

３章では，C 言語プログラムを構文木化する手法を提案している．まずは C 言語ソースプログラムに対する前処理を行う．次に，C 言語プログラムの構成パターンを分類し，それぞれのパターンを表現する構文木部品を作成する．更に，これらの部品を用いて C 言語プログラム全体を一つの木グラフに変換する．論文では，提案した構文木化の方法を用いて作成した CCX ソフトの説明や，具体例を用いた変換結果の紹介も行っている．

４章では，構文木グラフの類似度を求める手法を提案している．既に提案された Tree Overlapping 手法を紹介した上で，すべての共通部分木を用いて構文木の類似度を測る New Tree Overlapping（NTO）方法を提案している．また，木グラフの深さを基準にノード間の最大マッチングを計算することによって構文木の類似度を測る Depth Matching（DM）手法を提案している．

５章では，サンプルプログラムを用いて提案した二つの手法と既存の SMMT 手法との比較実験を行い，その結果について議論している．サンプルプログラムは，学生のレポートやテキストに掲載のプログラムや研究用のプログラムなどである．実験結果から，提案の DM 手法が最も良く，その有効性が確認できた．

６章では，本論文で得られた結果をまとめており，今後の展望や課題について述べている．

以上の論文の内容から，審査委員会は以下のように判断した．

１．創造性について

本研究の提案手法は C 言語プログラムを根付き順序木に変換し，変換された木グラフに対して類似性を定量的に評価するものである．提案の New Tree Overlapping（NTO）と Depth Matching（DM）の方法は，プログラムの実行文のみならず，プログラムの構造も比較の対象となっている．この技法は先行研究に見られないことで，創造性において優れている．

２．論理性について
本論文では，C 言語プログラムの木グラフへの変換や New Tree Overlapping（NTO）と Depth Matching（DM）の計算を論理的に行っており，これらの記述についても厳密に書かれていることから，論文全体として論理性において優れている．
３．厳格性について
先行研究を調査した上，プログラム構造の類似性定量評価を課題に設定している．その課題を解決するための提案手法の評価については，実際学生が提出したレポートを含む複数のサンプルプログラムを用いて行っており，既存の手法との比較・検討も行っている．これらのことから，厳格性において十分に達成できている．
４．発展性について
本論文で提案された類似性評価方法はプログラミング教育現場で活用できるだけでなく，将来的にシステム開発現場におけるソフトウェア変更の履歴やバージョン管理・検証への応用につながる．発展性においても十分に達成できている．
以上のように，創造性と論理性は「優れている」，厳格性と発展性は「達成できている」との評価であることから，全体評価としては「達成できている」と判断した．よって，審査委員会は論文審査結果を合と判定した．

論文審査結果
㊗・否

審査委員　主　査　（氏 名）　葛　崎偉

（氏 名）　成富　敬

（氏 名）　福田　隆眞

（氏 名）　山口　慎吾

（氏 名）　　㊞