4C-6

トランザクショナル記号実行

荒堀　喜貴†　　横田　治夫†
†東京工業大学大学院情報理工学研究科

1　はじめに

ソフトウェアの信頼性や性能の不具合解析に利用される動的記号実行（DSE）は，モデル検査などの静的解析手法に比べ正確かつスケーラブルであるという利点を持つ．しかし，大規模分散データ処理系のような並列分散系を対象とする場合，従来の DSE は二つの重大な問題に直面する．第一に，並列分散計算の非決定性により不具合が発現する状態（目的状態）の特定が困難になる．第二に，目的状態とその再現に必要な入力を運良く特定できた場合でさえ，同一入力を与えるだけの単純な再実行では目的状態を正確かつ高速に再現できない．

我々はこれらの問題の解決に向け，トランザクショナル記号実行と呼ぶ新たな動的解析方式を提案する．本方式は並列分散データ工学におけるトランザクション処理（TP）の要素技術に基づき DSE の枠組みを拡張したものである．具体的には，DSE における制約処理を TP における競合解析技術で拡張することにより，目的状態再現のための実効的パス・インタリーブ制約の合成を可能にする．更に，TP のロールバック制御に基づく有効範囲限定型の再実行により，DSE の目的状態の正確かつ効率的な再現を可能にする．

本発表では，提案方式の概要を紹介し，既存手法との定性的比較を議論する．

2　動的記号実行

2.1　基本的枠組み

動的記号実行（DSE）[1, 2, 3] は，プログラムの信頼性や性能の解析手法の一種であり，対象コードを目的状態に誘導する入力データを実行時情報から計算する．目的状態とは，例えば，プログラムが管理する機密情報が誤って外部に送信されるなどセキュリティ上問題のある状態やプログラムの実行が所望の期限内に完了しないなど性能上問題のある状態である．

DSE では，対象プログラムの実行と同時にパス制約と呼ばれる制約を収集する．パス制約は（1）プログラムへの入力データを表現する記号と（2）プログラム実行中に通過した各分岐文の分岐成立条件からなる論理式である．対象コードを目的状態に遷移させるために，DSE は収集したパス制約中の特定の分岐成立条件を改変することで，「他方向の分岐」を実現する新たなパス制約を生成する．生成したパス制約を SAT/SMT などの制約ソルバで解き，得られた充足解（具体的な入力データ）をプログラムに与えて再実行する．この再実行で，対象コードは実際に狙った方向に分岐する．

このように，DSE は（1）対象コード実行時のパス制約の収集，（2）他方向分岐パス制約の生成と充足，（3）充足解に基づくプログラム再実行という三つのステップをプログラム実行が所望の状態に遷移するか所定のコード網羅率を達成するまで繰り返す．所定のコード網羅率の達成まで問題とする状態が顕在化しなかった場合は，その事実をプログラムの品質の証とする．

2.2　問題

従来の DSE は，逐次実行コードに対しては有効であるが，大規模分散データ処理系のような並列分散プログラムを対象とする場合，次の問題点に直面する．

- **問題点 1:** 並列分散計算の非決定性により，目的状態の特定が困難である．
- **問題点 2:** 目的状態の再現に必要な入力を与えるだけの単純な再実行では目的状態を再現できない．

並列分散プログラムの非決定性とは，同じ入力を与えて実行しても同じ結果が得られるとは限らないという性質である．この性質は，並列分散処理を構成するスレッド群の各操作の実行順序（スレッドインタリーブ）が実行のたびに異なることに起因する．並列分散プログラムの実行では，逐次コードで問題となる不具合に加え，競合やデッドロックや順序違反など稀なスレッドインタリーブでしか顕在化しない不具合が問題となる．しかし，従来の DSE によるパス制約の生成はスレッドインタリーブを考慮しておらず，稀なインタリーブでのみ発現する目的状態の特定が困難である．

また，仮に目的状態とその再現に必要な入力を特定できたとしても，同一入力を与えてプログラムを再実行するだけでは特定のスレッドインタリーブを再現できず，目的状態に正確かつ効率的に到達できない．

†Yoshitaka ARAHORI　†Haruo YOKOTA
†Graduate School of Information Science and Engineering, Tokyo Institute of Technology

3 提案方式

3.1 実効的パス・インタリーブ制約合成

並列分散計算の非決定性に対応する目的状態の特定法として，実効的パス・インタリーブ制約合成を提案する．この手法は，DSE のパス制約の生成をトランザクション処理（TP）の競合解析技術で拡張することで，目的状態再現のためのパス制約とインタリーブ制約の合成を可能にする．拡張の要点は以下の通り．

- **要点 1:** DSE の各パス制約に対し，制約中の共有資源アクセス間の依存関係とインタリーブ可能性を記録し，拡張競合グラフとして表現．
- **要点 2:** 拡張競合グラフ上での到達可能性判定に基づき，目的状態再現のためのパス・インタリーブ制約を合成．

拡張競合グラフは，各スレッドの共有資源アクセスとアクセス時のパス制約をノードとし，共有資源アクセス間の依存関係とインタリーブ可能性を表現する二種類のエッジを持つ．このグラフ上で，あるスレッド T_1 による共有資源 S のアクセス $A_1(S)$ から他のスレッド T_2 による競合アクセス $B_2(S)$ に到達可能であれば，到達経路上にノードまたはエッジとして出現するパス制約とインタリーブ可能性の列を目的状態再現のための実効的パス・インタリーブ制約とする．

3.2 目的状態の正確かつ効率的な再現

合成によって得られた実効的パス・インタリーブ制約は，通常の DSE のパス制約に加え，各スレッドからの共有資源アクセスの実行順序をインタリーブ制約として保持する．パス制約の充足解を入力データとしてプログラムを再実行し，インタリーブ制約の充足解に基づき再実行中のスレッドインタリーブを制御することにより，非決定的な並列分散計算の目的状態を正確に再現することができる．

しかし，この再現法は従来の DSE と同様，プログラムを最初から再実行するため効率が悪い．そこで，この問題への対処法として，TP のロールバック制御に基づく有効範囲限定型の再実行方式を提案する．この方式の要点は次の通り．

- **要点 1:** DSE において，各スレッドの共有資源へのアクセス要求列をトランザクションとして管理．
- **要点 2:** 目的状態再現のために，必要なスレッド群を犠牲者（victim）として選択した上で各々を必要な時点までロールバック．

まず，DSE の実行時に，観測対象の各スレッドからの共有資源アクセス要求列をトランザクション [4] として管理し，実効的パス・インタリーブ制約の合成（による目的状態の特定）が完了するまでコミットを待つ．合成完了後，得られたインタリーブ制約と DSE が観測した共有資源アクセス実行順序の整合性を調べる．ここで，矛盾があった場合，インタリーブ制約の充足に必要なスレッド群を犠牲者として選択する．例えば，共有資源アクセス列 $A_1(S)...B_2(S)$ の観測に対し，インタリーブ制約が $B_2(S) \rightarrow A_1(S)$（観測と逆順）である場合，スレッド T_1 を $A_1(S)$ 実行直前の時点までロールバックすることにより目的状態を再現する．

4 議論

従来の DSE[1, 2, 3] が考慮しないインタリーブ制約の合成を競合グラフの到達可能性判定に帰着させることにより，我々の提案手法は並列分散計算の非決定性に対し目的状態の正確な特定を可能にする．更に，ロールバック制御に基づく再実行は，必要なスレッドを途中から再実行するだけで済むため，従来の DSE に比べ目的状態の再現が高速である．

5 おわりに

従来の動的記号実行の問題点をトランザクション処理の要素技術で解決する手法を提案し，その機能について関連手法との定性的比較を議論した．今後，提案手法の高度化に向け，静的/動的混成式のプログラム解析に基づく（1）共有資源の正確な識別法や（2）最適なトランザクション粒度の決定法を実現する．

参考文献

[1] Sen, K., Marinov, D., and Agha, G.: CUTE: A Concolic Unit Testing Engine for C, In *FSE*, pp. 263–272 (2005).

[2] Godefroid, P., Klarlund, N., and Sen, K.: DART: Directed Automated Random Testing, In *PLDI*, pp. 213–223 (2005).

[3] Cadar, C., Dunbar, D., and Engler, D.: KLEE: Unassisted and Automatic Generation of High-coverage Tests for Complex Systems Programs, In *OSDI*, pp. 209–224 (2008).

[4] Shavit, N. and Touitou, D.: Software Transactional Memory, In *PODC*, pp. 204–213 (1995).