

Analyse Paralleler Programme durch Kombination Automatenbasierter Erreichbarkeitsanalyse mit Datenflussanalyse

Extended Abstract

Alexander Wenner

Institut für Informatik, Fachbereich Mathematik und Informatik
Westfälische Wilhelms-Universität Münster
alexander.wenner@uni-muenster.de

Die Entwicklung paralleler Programme hat in den vergangenen Jahren stark zugenommen. Dazu haben insbesondere das Aufkommen von Programmiersprachen die Parallelität direkt unterstützen und die zunehmenden Verfügbarkeit paralleler Hardware beigetragen. Parallele Programmierung ist jedoch komplex und daher fehleranfällig. Statische Analysen können den Programmierer zur Entwicklungszeit auf mögliche Fehlerquellen aufmerksam machen und ihn damit bei der Entwicklung unterstützen.

In den letzten Jahren wurden Erreichbarkeitsprobleme für Varianten dynamischer Pushdown Netzwerke (DPN) [1], als abstraktes Modell für den Kontrollfluss paralleler rekursiver Programme, untersucht. DPN sind eine Erweiterung von Kellerautomaten, die mehrere parallel arbeitende Stacks modellieren, deren Anzahl sich zu Laufzeit dynamisch ändern kann. Im Gegensatz zu der Unentscheidbarkeit von Erreichbarkeit für starke Synchronisationsprimitive [2] bleibt Entscheidbarkeit erhalten wenn Synchronisation durch well-nested Locking [3] realisiert wird. Dies kann um zusätzliche einfache Joins erweitert werden [4]. Alternativ ist Erreichbarkeit auch für contextual Locking und einfache Joins entscheidbar [5].

Wir erweitern zunächst das Modell aus [4] um ausdrückstärkere Joins und zeigen, dass Erreichbarkeit präzise entscheidbar bleibt. Hierzu adaptieren wir die automatenbasierten Techniken aus [4] und zeigen, dass die Menge kritischer Ausführungen durch eine reguläre Menge von Bäumen dargestellt werden können. Erreichbarkeit kann damit durch den Leerheitstest eines Baumautomaten überprüft werden.

In einem zweiten Schritt präsentieren wir eine Möglichkeit, unter Verwendung des obigen Resultats, Datenflussanalysen für sequentielle Programme auf parallele Programme zu erweitern. Verschiedene existierende Ansätze [6,7,8] basieren auf der Idee einzelne Prozesse in einem parallelen Programm zu betrachten und interferierende Operationen paralleler Prozesse als zusätzliche Operationen in die Datenflussanalyse einfließen zu lassen. Aufbauend auf dieser Idee entwickeln wir ein Framework für Datenflussanalysen in dem wir die Erreichbarkeitsanalyse für DPN verwenden um die Interferenz zwischen Prozessen in parallelen Programmen zu approximieren. Zu diesem Zweck konstruieren wir eine Folge von Abstraktionen, im Kontext von Abstrakter Interpretation [9], die es uns erlau-

ben einzelne sequentielle Prozesse zu betrachten und gleichzeitig ein DPN zu extrahieren mit dem wir Interferenzen extrahieren können.

In Kombination erhalten wir eine Vorgehensweise um Datenflussanalysen für sequentielle Programme auf parallele Programme zu erweitern.

Literatur

1. Bouajjani, A., Müller-Olm, M., Touili, T.: Regular symbolic analysis of dynamic networks of pushdown systems. In: CONCUR 2005. LNCS 3653, Springer (2005)
2. Ramalingam, G.: Context-sensitive synchronization-sensitive analysis is undecidable. *ACM Trans. Program. Lang. Syst.* **22**(2) (2000)
3. Lammich, P., Müller-Olm, M., Wenner, A.: Predecessor sets of dynamic pushdown networks with tree-regular constraints. In: CAV 2009. Volume 5643 of Lecture Notes in Computer Science., Springer (2009) 525–539
4. Gawlitza, T.M., Lammich, P., Müller-Olm, M., Seidl, H., Wenner, A.: Join-lock-sensitive forward reachability analysis for concurrent programs with dynamic process creation. In: VMCAI 2011. Volume 6538 of Lecture Notes in Computer Science., Springer (2011) 199–213
5. Lammich, P., Müller-Olm, M., Seidl, H., Wenner, A.: Contextual locking for dynamic pushdown networks. In Logozzo, F., Fähndrich, M., eds.: *Static Analysis*. Volume 7935 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 477–498
6. Miné, A.: Static analysis of run-time errors in embedded critical parallel C programs. In: Proc. of the 20th European Symposium on Programming (ESOP'11). Volume 6602 of Lecture Notes in Computer Science (LNCS)., Springer (Mar. 2011) 398–418 <http://www.di.ens.fr/~mine/publi/article-mine-esop11.pdf>.
7. Jeannet, B.: Relational interprocedural verification of concurrent programs. *Software and System Modeling* **12**(2) (2013) 285–306
8. Seidl, H., Vene, V., Müller-Olm, M.: Global invariants for analyzing multithreaded applications. *Proc. of the Estonian Academy of Sciences: Phys., Math.* **52**(4) (2003)
9. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: POPL, ACM (1977)