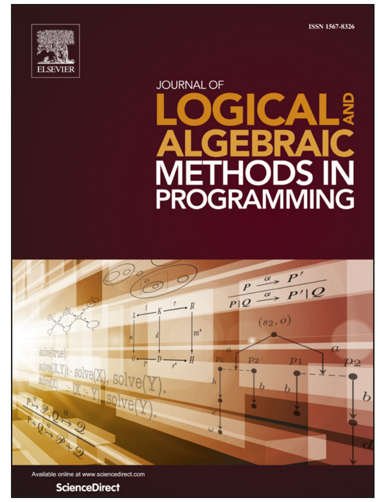
Accepted Manuscript



Formalization of Habanero Phasers using Coq

Tiago Cogumbreiro, Jun Shirako, Vivek Sarkar

PII: S2352-2208(16)30083-9
DOI: http://dx.doi.org/10.1016/j.jlamp.2017.02.006
Reference: JLAMP 169

To appear in: *Journal of Logical and Algebraic Methods in Programming*

Received date: 10 August 2016
Revised date: 5 January 2017
Accepted date: 22 February 2017

Please cite this article in press as: T. Cogumbreiro et al., Formalization of Habanero Phasers using Coq, *J. Log. Algebraic Methods Program.* (2017), http://dx.doi.org/10.1016/j.jlamp.2017.02.006

Journal of Logical and Algebraic Methods in Programming

Available online at www.sciencedirect.com
ScienceDirect

Journal of Logical and Algebraic Methods in Programming 00 (2017) 1–16

Formalization of Habanero Phasers using Coq

Tiago Cogumbreiro, Jun Shirako, Vivek Sarkar

Rice University, 6100 Main St, Houston, TX 77005, USA

Abstract

Phasers pose an interesting synchronization mechanism that generalizes many collective synchronization patterns seen in parallel programming languages, including barriers, clocks, and point-to-point synchronization using latches or semaphores. This work characterizes scheduling constraints on phaser operations, by relating the execution state of two tasks that operate on the same phaser. We propose a formalization of Habanero phasers, May-Happen-In-Parallel, and Happens-Before relations for phaser operations, and show that these relations conform with the semantics. Our formalization and proofs are fully mechanized using the Coq proof assistant, and are available online.

Keywords: phasers, barriers, Coq, synchronization, formalization

1. Introduction

Phasers are an interesting synchronization mechanism that generalizes barriers with collective producer-consumer synchronization. A phaser can encode the synchronization mechanism of latches, futures, join barriers, cyclic barriers, as well as any *collective synchronization pattern* provided by CUDA, C♯, Java, MPI, and X10. Phasers [1] were first introduced in the Habanero Extreme Scale research project at Rice University, as an extension to X10 clocks [2], and implemented in Habanero-Java and Habanero-C. A restricted form of phasers was also introduced in the standard `java.util.concurrent.Phaser` library starting with Java 7. The phaser synchronization mechanism is relevant at the theoretical level because of its generality. Theoretical results that target phasers can easily translate across different languages and parallel runtimes [3].

This paper introduces the first formalization of: (1) the Habanero phaser semantics, and (2) the *Phase Ordering* relation proposed in [1], which is a relation used for causality analysis [4; 5]. The former is a crucial stepping stone for formal developments focused in many-to-many synchronization. The latter gives us an interpretation of a phaser as a logical clock. Causality analysis are fundamental in the verification of barrier synchronization errors [6], lock-based deadlock prediction [7; 8], and race-detection [9; 10].

Email addresses: `cogumbreiro@rice.edu` (Tiago Cogumbreiro), `shirako@rice.edu` (Jun Shirako), `vsarkar@rice.edu` (Vivek Sarkar)