Classification of Dataflow Actors with Satisfiability and Abstract Interpretation

Matthieu Wipliez, IETR/INSA, France
Mickaël Raulet, IETR/INSA, France

ABSTRACT

Dataflow programming has been used to describe signal processing applications for many years, traditionally with cyclo-static dataflow (CSDF) or synchronous dataflow (SDF) models that restrict expressive power in favor of compile-time analysis and predictability. More recently, dynamic dataflow is being used for the description of multimedia video standards as promoted by the RVC standard (ISO/IEC 23001:4). Dynamic dataflow is not restricted with respect to expressive power, but it does require runtime scheduling in the general case, which may be costly to perform on software. The authors presented in a previous paper a method to automatically classify actors of a dynamic dataflow program within more restrictive dataflow models when possible, along with a method to transform the actors classified as static to improve execution speed by reducing the number of FIFO accesses (Wipliez & Raulet, 2010). This paper presents an extension of the classification method using satisfiability solving, and details the precise semantics used for the abstract interpretation of actors. The extended classification is able to classify more actors than what could previously be achieved.

Keywords: Abstract Interpretation, Classification, Dataflow Program, Satisfiability, Signal Processing

INTRODUCTION

The arrival of multi-core in the desktop computing market has renewed the interest in multi-processor programming. There are many different techniques to write multiprocessing programs, depending on memory architecture (shared or distributed), processor architecture (uniprocessor or multiprocessor), number of cores (single core, multi-core, many-core), etc. Most of these techniques constrain program design in a way that makes it difficult to use a different technique, should the program be ported to a different architecture than initially planned. Dataflow programming is a portable platform-agnostic alternative that allows an algorithm to be described so that parallelism is made explicit.

A dataflow description is a directed graph where vertices (or actors) process data and edges carry data, with the requirement that vertices cannot share data. Actors can only communicate with other actors through ports connected to edges. The semantics of a dataflow program...
are defined by a Model of Computation (MoC) that dictates conditions for existence of a valid schedule, bounded memory consumption, proof of termination, and other properties. MoCs go from Synchronous Dataflow (SDF) with total compile-time predictability with respect to scheduling, memory consumption, termination, to dynamic dataflow where those properties are not predictable in the general case, with increasing expressiveness, for instance see Lee and Messerschmitt (1987), Bilsen, Engels, Lauwereins, and Peperstraete (1996), Bhattacharya and Bhattacharyya (2001), Buck and Lee (1993), Buck (1994), and Lee and Parks (1995).

The Reconfigurable Video Coding (RVC) standard defines existing MPEG video standards as dynamic dataflow programs in which actors are written in a language called RVC-CAL (Mattavelli, Amer, & Raulet, 2010). These dataflow programs can be automatically translated and ported to a wide range of platforms and languages, from hardware to multi-core software (Mattavelli et al., 2010). Contrary to SDF and a few other MoCs, RVC-CAL has a much greater expressive power, but this also means that runtime scheduling is mandatory in the general case, which impacts execution speed.

Fortunately, most signal processing applications are far from being entirely dynamic, and parts with static behavior need not be dynamically scheduled. The problem is to detect actors that behave statically or quasi-statically, since dynamic dataflow has an expressive power equivalent to a Turing machine (Buck & Lee, 1993), which means it is not possible to prove the termination of a dynamic dataflow program in general. We presented in a previous paper a method to automatically classify actors of a dynamic dataflow program within more restrictive dataflow models when possible, along with a method to transform the blocks classified as static to improve execution speed by reducing the number of FIFO accesses (Wipliez & Raulet, 2010).

This paper makes the following contributions:

- We describe an Intermediate Representation (IR) of dataflow actors that is more suitable for classification than the Abstract Syntax Tree (AST) of actors, and how to translate the AST of an actor to its IR.
- We describe the precise semantics behind the abstract interpretation of the IR of an actor, which is the basis for our classification method that is able to discover facts about an actor without the need for actual data. We show an example of how abstract interpretation can be used to find cyclo-static behavior.
- We present the extension of our classification method that is able to determine more precisely actors that present a time-dependent behavior. Time-dependent behavior can be used for low-level optimizations, but it makes the actor behave in a non-deterministic way (it is possible, however, for an application with some nondeterministic actors to behave in a deterministic way). Because the RVC standard specifies the behavior of video standards with RVC-CAL dataflow actors, and the specification is expected to be deterministic, it is vital to be able to prove that a given actor is deterministic.
- We show how the IR of a dataflow actor is translated to the standard format used for satisfiability solving called SMT-LIB. SMT (Satisfiability Modulo Theories) allows us to determine properties of the source code that are required by our classification method.

**BACKGROUND**

**A Taxonomy of Dataflow Models of Computation**

A Model of Computation (MoC) defines the behavior of a program described as a dataflow graph. A dataflow graph is a directed graph whose vertices are actors and edges are unidirectional FIFO channels with unbounded capacity, connected between ports of actors. Dataflow
Knowing the Enemy at the Gates: Measuring Attacker Motivation
www.igi-global.com/article/knowing-the-enemy-at-the-gates/79283?camid=4v1a