

# A Programmer's Guide to AspectLTL

Version 1.0, November 2010

Shahar Maoz<sup>1</sup> and Yaniv Sa'ar<sup>2</sup>

<sup>1</sup> RWTH Aachen University, Germany  
maoz@se-rwth.de

<sup>2</sup> The Weizmann Institute of Science, Israel  
yaniv.saar@weizmann.ac.il

**Abstract.** This document is a programmer's guide to AspectLTL [2], a temporal-logic based language for the specification and implementation of crosscutting concerns. The guide provides an overview of AspectLTL from a programmer's perspective, covering the main constructs and features of the language using a number of examples, together with an overview of the main features of the AspectLTL plug-in.

## 1 Introduction

### 1.1 What is AspectLTL?

AspectLTL [2] is a language for the specification and implementation of crosscutting concerns, based on linear temporal logic (LTL) [4]. The aspects of AspectLTL, called LTL aspects, enable the declarative specification of expressive crosscutting concerns. These include the specification of safety properties, which may be used to prevent a base system from visiting 'bad states', the specification of liveness (fairness) properties, which may be used to force a base system to visit 'good states' (infinitely often), and the addition of new behaviors to a base system, which is done by specifying the existence of new transitions and new states as necessary.

AspectLTL comes with a synthesis-based weaving process, whose output is a correct-by-construction executable artifact. Following a composition of the specified aspects with a base system, using symbolic disjunctive and conjunctive operations, we formulate the problem of correct weaving as a synthesis problem [5], essentially a game between the environment and the (augmented) base system. An algorithm based on [3] is used to solve the game, that is, to provide the augmented system with a winning strategy, if any.

If a winning strategy is found, it is presented as a deterministic, executable automaton, which represents an augmented base system whose behavior is guaranteed to adhere to the specified aspects, in all possible environments. If a winning strategy is not found, we know that it does not exist, that is, that no system exists which is based on the base system and can adhere to the specified LTL aspects in all environments. Thus, LTL aspect composition and synthesis is sound and complete.