A Programming Language for Origami Construction and Theorem Proving

Fadoua Ghourabi* and Tetsuo Ida

Department of Computer Science
University of Tsukuba
Tsukuba 305-8573, Japan
{ghourabi, ida}@score.cs.tsukuba.ac.jp
phone:+81 29-853-5164/fax:+81 29-853-6982

Abstract

Recipes for origami are traditionally described in combination of a natural language and diagrams. While this is sufficient for usual practice of origami, formal ways of prescribing origami constructions are needed in technological treatment of origami, e.g. in origami geometry and in origami databases. We present a formal programming language for origami construction and verification, which is currently focussed on origami 2D geometry.

In origami geometry, we have two phases of reasoning; reason about the construction and about subsequent correctness proof of the construction. During the construction phase, we are concerned with finding fold lines. This requires the description of constructed points and segments as the parameters that determine the fold lines. The language that we need during the construction is the one that can define and manage the constructed objects together with the basic origami operations such as fold, unfold and marking constructed points. In the proof phase, we need to specify the geometrical properties that should be proven. Moreover, the language should allow us to give stepwise description of origami constructions and communicate with the computer system interactively. The language is in essence the oner tuned for geometrical objects and is based on the fragment of the first-order predicate logic with build-in functions interpreted over the algebraic fields. It has supports of maintaining the geometrical objects and modular constructions of origamis.

The interpreter of the language has two kinds of engines, i.e. constraint solvers and provers based on symbolic computation methods such as Groebner basis and cylindrical algebraic decomposition. When one piece of origami construction and verification is completed, the interpreter produces a proof document describing how the origami is constructed, algebraically treated and proven. The language is implemented on top of Mathematica and is the interface of our computational origami system Eos.

*Corresponding author.