LoCo - A Logic for Configuration Problems

Markus Aschinger

St Anne's College, University of Oxford

Doctor of Philosophy, Hilary Term 2014

Abstract

This thesis deals with the problem of technical product configuration: Connect individual components conforming to a component catalogue in order to meet a given objective while respecting certain constraints. Solving such configuration problems is one of the major success stories of applied AI research: In industrial environments they support the configuration of complex products and, compared to manual processes, help to reduce error rates and increase throughput. Practical applications are nowadays ubiquitous and range from configurable cars to the configuration of telephone communication switching units.

In the classical definition of a configuration problem the number of components to be used is fixed while in practice, however, the number of components needed is often not easily stated beforehand. Existing knowledge representation (KR) formalisms expressive enough to deal with this dynamic aspect of configuration require that explicit bounds on all generated components are given as well as extensive knowledge about the underlying solving algorithms. To date there is still a lack of high-level KR tools being able to cope with these demands.

In this work we present LoCo, a fragment of classical first order logic that has been carefully tailored for expressing technical product configuration problems. The core feature of LoCo is that the number of components used in configurations does not have to be finitely bounded explicitly, but instead is bounded implicitly through the axioms. We identify configurations with models of the logic; hence, configuration finding becomes model finding. LoCo serves as a high-level representation language which allows the modelling of general configuration problems in an intuitive and declarative way without the need of having knowledge about underlying solving algorithms; in fact, the specification gets automatically translated into low-level executable code. LoCo allows translations into different target languages. We present the language, related algorithms and complexity results as well as a prototypical implementation via answer-set programming.