Student Project—Newton-Solver for Semirings

Maximilian Schlund

January 25, 2012

Many program analysis tasks can be reduced to solving a system of polynomial equations $X = f(X)$ over a semi-ring. The goal of this project is to implement Newton's method for semi-rings as described in [1]. Possible applications are verification and program analysis.

1 Project Tasks

Compositional construction and manipulation of semirings. This package is motivated by static analyzers like *AbsInt*[1]. In AbsInt the complete lattice to be used for the abstract interpretation can be specified by the user by combining basic lattices, e.g., two power-set algebras on disjoint finite sets can be combined into their product lattice.

In a similar fashion, we plan to give the user the possibility to specify a ω-continuous semiring by building up on basic classes of ω-continuous semirings like counting semirings on $2^{\mathbb{N}^k}$ or the nonnegative real numbers. This work package therefore consists of both defining and implementing a user-friendly specification language for building complex semirings and also the study of how the solvers for the basic semirings can be efficiently combined to solvers for composite semirings.

Algorithms and data structures for semirings. In this work package a library will be implemented which allows to handle the most basic types of ω-continuous semirings for which the Kleene star can be computed. These are semirings with finite carrier, the semiring of the nonnegative reals, and commutative idempotent (aka. counting) and aperiodic semirings. In a first step, algorithms and data-structures for the different semirings have to be designed and evaluated. We plan to start with the real semiring as we have already gathered a good working knowledge by our research on the convergence speed of Newton's method.

The representation of finite semirings ultimately depends on the specification given by the user. The main body of work for finite semirings is therefore the specification of a user-friendly interface which then can be used within the preceding work package.

[1] `http://www.absint.com/`

Finally, the representation of the commutative idempotent and aperiodic semirings boils down to representing semilinear sets. Although semilinear sets of fixed dimension are of undeniable importance for computer science, there seems to be no widely used implementation of a library for handling semilinear sets: Several possible representations are known for semilinear sets, e.g., regular expressions, finite automata, specifically number decision diagrams. We have already gathered some preliminary knowledge on handling of semilinear sets using finite automata[2]. As there is no best representation of semilinear sets in general, we will give the user the freedom of choice. This will also allow us to compare and evaluate the currently known approaches for handling semilinear sets.

Another interesting task is to investigate the representation of semilinear sets as finite sums of rational generating functions for the lattice cones ("linear sets") (see the work of de Loera, Hemmecke, Sturmfels,...). There is a polynomial (for fixed dimension!) encoding algorithm (Barvinok's Algorithm) to derive a short rational generating function for a lattice cone (see also the tool LattE by de Loera, Hemmecke, et. al.).

2 Summary

- Implement a framework that allows for an easy definition and efficient internal representation of semirings (general, commutative, idempotent, aperiodic,. . .). Implement also some concrete instances (real semiring, counting semiring).
- Implement Newton's method for abstract semirings (commutative, idempotent —e.g. parametrized by a method to compute the Kleene star). Instantiate the framework with some concrete examples. Experiment!

This project can be pursued as a Bachelor's or Master's thesis, as an IDP, as a guided research, or as a "HIWI"- project, depending on your time, previous knowledge, and interests.

3 Contact

If you are interested please write an email to Maximilian Schlund (schlund@model.in.tum.de)or just drop by at my office (Room 03.11.055).

References

[1] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian program analysis. Journal of the ACM, 57(6):33:1–33:47, October 2010.

[2] `http://www.fmi.uni-stuttgart.de/szs/tools/bnddwpds/`