INFINITY

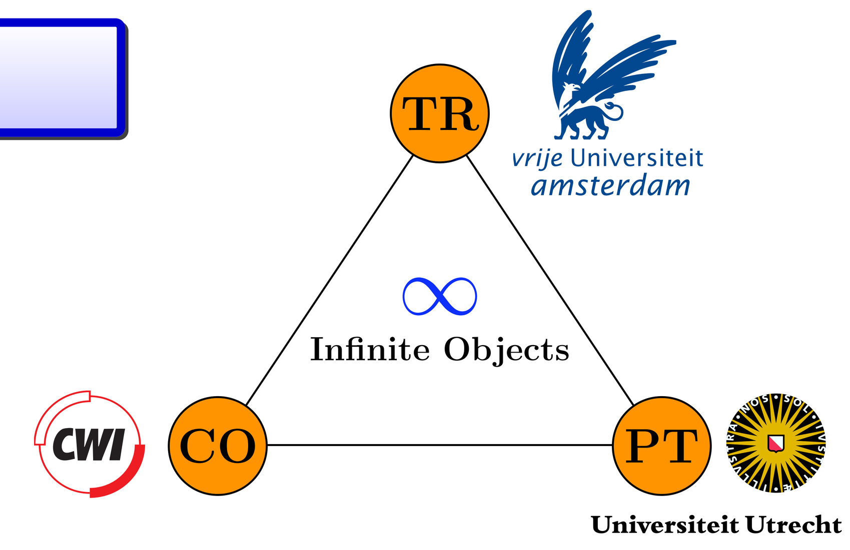
Modelling, Computing, and Reasoning with Infinite Objects

Clemens Kupke, Dimitri Hendriks, and Clemens Grabmayer

CWI and VU, Amsterdam, UU, Utrecht, the Netherlands

The Project

In computer science recently much attention has shifted from finite data to infinite data. In this project infinite objects are studied from the perspectives of Coalgebra (CWI, Clemens Kupke, Helle Hvid Hansen), Term Rewriting (VU, Dimitri Hendriks, Ariya Isihara, Jörg Endrullis), and Proof Theory (UU, Clemens Grabmayer). The goal is to integrate these approaches, and to develop new techniques for modelling, computing, and reasoning with infinite objects. Areas of primary interest in the project have been up to now: the theory of stream specifications, and coalgebraic logic.

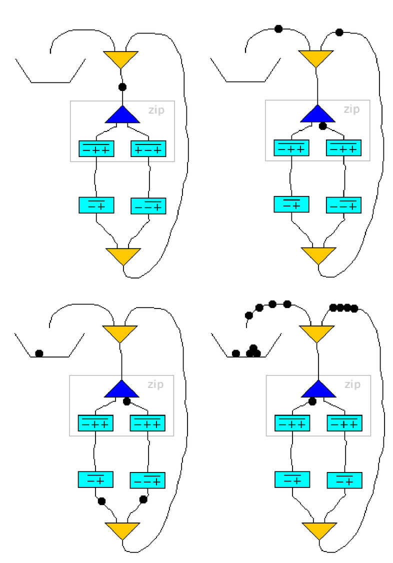


Productivity of Stream Specifications

A stream specification is called *productive* if it can be evaluated continually in such a way that a uniquely determined stream is obtained as the limit.

Whereas productivity is undecidable for stream specifications in general, we established that it can be decided for *pure* stream specifications, a large and natural class of recursive stream specifications (see [1]). For every pure stream specification the process of its evaluation can be modelled by the dataflow of abstract stream elements, called *pebbles*, in a finite *pebbleflow net*. And the production of a pebbleflow net associated with a pure stream specification, that is, the amount of pebbles the net is able to produce at its output port, can be calculated by reducing nets to trivial nets.

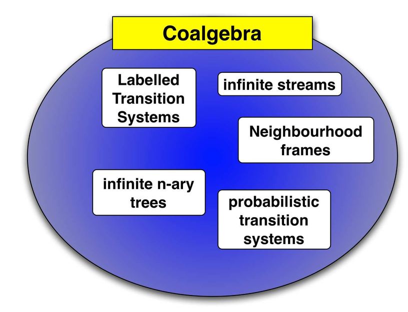
See right for pictures illustrating four states of the productive pebbleflow net associated with the pure stream specification $\mathsf{M} = 0 : \mathsf{zip}(\mathsf{i}(\mathsf{M}), \mathsf{tail}(\mathsf{M}))$ of the Thue–Morse sequence $0:1:1:0:1:0:0:1:1:0:0:1:0:1:1:0:\ldots$.



Coalgebraic Logics and Their Semantics

Coalgebras provide a framework for studying various types of dynamical and transition systems in a uniform manner, e.g. labelled transitions systems, infinite n-ary trees and infinite words are examples of coalgebras. Coalgebraic logics are a family of modal logics that are used to reason about coalgebras.

From a logical perspective, neighbourhood frames are a fundamental example: the coalgebraic modal logic of neighbourhood frames is the so-called "classical" modal logic, i.e. the modal logic in which the modal operators satisfy the congruence rule and no extra axioms. In [2] we introduced a notion of bisimilarity between neighbourhood frames and proved an analogue of Van Benthem's theorem: over neighbourhood frames classical modal logic is the bisimulation invariant fragment of first-order logic.



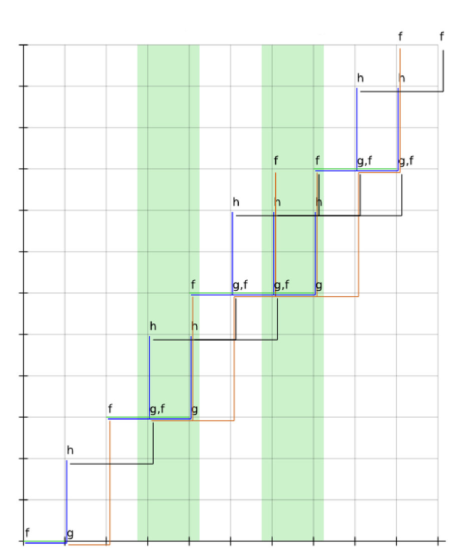
Criteria for (Non-)Productivity by a Data-Oblivious Analysis

We extend the above decidability result for productivity:

(1) We formulate computable sufficient conditions for (non-)productivity of stream specifications under the assumption that *rational* lower (upper) bounds for the production of all stream functions occurring are given.

(2) For an extension of pure stream specifications employing exhaustive data pattern analysis, we give a computable criterium for productivity that is provably optimal among all *data-oblivious* approaches (in which all knowledge about concrete values of data elements is ignored).

For (2) we employ an algorithm that computes optimal data-oblivious rational lower bounds for stream function specifications involving pattern matching on data. This algorithm develops all relevant data-oblivious consumption/production traces until a repetition is detected (see the green strips in the picture right), from which the lower bound can be extracted.



Stream Definitions in Different Cobases

A given stream $\sigma \in \mathbb{N}^\omega$ can be described with different operations. For example we can define a stream σ to be an element $\mathsf{head}(\sigma) \in \mathbb{N}$ followed by another stream $\mathsf{tail}(\sigma)$ or, alternatively, we can describe σ by specifying its first element $\mathsf{head}(\sigma) \in \mathbb{N}$ and two streams $\mathsf{odd}(\sigma)$ and $\mathsf{even}(\sigma)$. In hidden algebra the notion of a cobase gives a formal criterion for when such a description is complete. We looked at cobases from a coalgebraic perspective and rephrased their definition in category-theoretic terms. Our aim is to devise and study a family of co-inductive stream definition schemes that is parametric in the choice of the cobase. Furthermore our category-theoretic formulation will allow for a straightforward generalization to other infinite objects such as infinite binary trees and bi-infinite streams.

$$\begin{array}{ccc} \mathbb{N}^\omega & \dashrightarrow & \mathbb{N}^{2^*} \\ \downarrow{\scriptstyle \langle \mathsf{head}, \mathsf{even}, \mathsf{odd} \rangle} & & \downarrow{\scriptstyle \langle \mathsf{head}, \mathsf{left}, \mathsf{right} \rangle} \\ \mathbb{N} \times \mathbb{N}^\omega \times \mathbb{N}^\omega & \dashrightarrow & \mathbb{N} \times \mathbb{N}^{2^*} \times \mathbb{N}^{2^*} \end{array}$$

Further Goals and Questions

- Can we obtain, for interesting classes of stream specifications, complete proof systems for equivalence?
- Can the results obtained for stream specifications be adapted to results concerning productivity and co-bases for other infinite objects like infinite n-ary trees or bi-infinite streams?
- Can we refine the pebbleflow semantics to account for the delay of evaluation of stream elements, in order to obtain a more genuine modelling of lazy evaluation of stream specifications?
- Can we obtain a complete proof calculus for Moss' coalgebraic logic? The language of this logic is defined uniformly for a large family of coalgebras (in particular for the above mentioned examples) and is known to be rich enough to distinguish non-bisimilar points.
- Our productivity results exploits the fact that pebbleflow nets can be viewed as a data-oblivious semantics for the evaluation of stream specifications. Can this idea be applied for obtaining statements that clarify the extent to which term graph rewriting systems can be viewed as a semantics for term rewriting systems?

Publications

[1] J. Endrullis, C. Grabmayer, D. Hendriks, A. Isihara, and J.W. Klop. Productivity of Stream Definitions. In *Proceedings of FCT 2007*, number 4639 in LNCS, pages 274–287. Springer, 2007.

[2] H.H. Hansen, C. Kupke, and E. Pacuit. Bisimulation for Neighbourhood Structures. In *Proceedings of CALCO 2007*, number 4624 in LNCS, pages 279–293. Springer, 2007.