Synthesis with string diagrams in Haskell
Izaak Meckler, University of Chicago

Introduction

More [10] is an editor plugin written by the author which can synthesize Haskell terms of a requested type as simple combinations of a base set of rewrite functions. Without attention, many syntactically distinct yet semantically equivalent terms will be synthesized, cluttering the results presented to the user.

To that end, we have developed and implemented an algorithm based on the geometry of string diagrams for eliminating redundant terms in synthesis results.

String Diagrams

String diagrams [11] are a graphical notation for a class of expressions (approximately given by the grammar at right) that intrinsically reflect certain equations holding between such expressions. As a result, we can partially de-duplicate synthesized terms by converting them to string diagrams and eliminating duplicates (now literally so) string diagrams. See Figure 4 for an example of the kind of duplicate terms which are eliminated.

We think of string diagrams as drawings of labelled graphs identified up to “smushing around” (more formally a certain equivalence relation). String diagrams are also connected to the perhaps better known “pasting diagrams” of category theory, as illustrated in Figure 5.

The relationship between string diagrams and terms (left column) and some equations of terms (right column) of the 22nd International Conference on Theorem Proving in Higher Order Logics, pp. 73-78. Springer-Verlag, 2009.

References


Efficacy of deduplication

Efficacy of deduplication

We can number of trials to Mori’s synthesis procedure with various query types and term size limit. In each trial, the number of top synthesized terms decreased after deduplication. The height of the plot at n-coordinate indicates the percent of trials for which the percent of results remaining after deduplication was at most n.

Running time

Each colored line corresponds to a type for which Mori was quoted to synthesize terms of size at most 5, 6 and 7. The running time is roughly exponential in the maximum permitted size of the term, which is somewhat surprising since Mori finds all such terms. The running times begin to become too long for interactive usage at maximum size 5, but are quite speedy at smaller sizes. To remedy this, we would like to try running search off after a threshold number of high quality terms has been found.

Usefulness of synthesized terms

Unfortunately we have not yet done a systematic evaluation of the usefulness of synthesized terms. Anecdotally, the synthesis results are occasionally useful, but is clear that many desirable terms do not fit the form which Mori synthesizes, and a good deal of further engineering work would be required to adapt it to suit them.

Appendix

At the time of writing, due to a bug, our implementation does not include any function with a type of the form

\[
\forall a. a \to F a
\]

in the base set of functions used for synthesis.

Acknowledgements

Thanks to Ravi Chugh for all the advice throughout the development of this research.