Finite Limits and Anti-Unification in Substitution Categories

Wolfram Kahl

McMaster University, Hamilton, Ontario, Canada, kahl@cas.mcmaster.ca

It is well-known that coequalisers and pushouts of substitutions correspond to solutions of unification problems, and therefore do not always exist. But how about equalisers and pullbacks? If the literature contains the answers, they are well-hidden — we provide explicit constructions and proofs, and in particular work out the details of categorial products in categories of substitution morphisms, for which the universal arrow construction corresponds exactly to anti-unification.

Substitutions occur in the formal study of syntactic systems, and are mappings from “variables” to “terms” (or “expressions”). Terms may contain variables, and “application” of substitution to terms produces terms again. Rydeheard and Stell [RS87] introduced (based on closely related ideas by Lawvere [Law63]) a categorical treatment of substitutions, where objects are sets considered as sets of variables, and morphisms from V_1 to V_2 are substitutions that map each element of V_1 to a term containing only variables from V_2. They say: “In this case variables are *localized*.” This is in contrast with most of the conventional literature on substitutions, were mostly a single global set of variables is assumed. For example, Eder [Ede85] investigates a “more general than” order on idempotent substitutions.

Rydeheard and Stell [RS87] used their category-theoretic formulation in particular for the construction of a unification algorithm, where unification is defined as coequaliser of substitutions. Since unification problems are not always solvable, coequalisers do not always exist in substitution categories, but since unification problems are important, coequalisers (and pushouts which correspond to unification problems with disjoint variable sets) have received much attention in the literature. However, I have not been able to find explicit statements about equalisers, pullbacks, or products in substitution categories. Since substitution categories are a concrete instance of Kleisli categories, Szigeti's study on limits and colimits in Kleisli categories using adjunctions [Szi83] is related, but, in his own words: “Mention must be made that these results are powerless in concrete instances.”

We provide concrete constructions and detailed proofs for equalisers, products, and pullbacks in substitution categories. In summary, the instances of basic category-theoretic concepts for the substitution category for a fixed signature take the following shapes, where well-known general facts are included in parentheses for completeness:

1. Epimorphisms are those substitutions where all target variables occur in the image.

2. Monomorphisms are those substitutions for which the image of any variable does not result from applying that substitution to any term different from that variable, as previously shown in [Kah15].
3. Equalisers of two substitutions in the substitution category always exist, and are the variable subset injections for the subset on which the two substitutions have the same images — these are the equalisers of the two substitutions in *Set*, but the proof of the universal property is substitution-specific.
4. (Coequalisers are most-general unifiers, and do not always exist.)
5. Products have as objects the sets of all pairs of not-equally-headed terms; the construction of the universal morphism is essentially anti-unification. Products of finite (variable) sets are in general infinite.
6. (Coproducts are inherited from *Set* as for every monad.)
7. Pullbacks always exist (since they can be obtained from product and equalisers). Pullbacks of substitutions between finite (variable) sets have finite pullback objects.
8. (Pushouts correspond to most-general unifiers of substitutions with disjoint range, and do not always exist.) Pushouts along injective variable renamings do exist.

We are working on a mechanization of this theory in the dependently-typed programming language and proof assistant Agda [Nor07]; at the time of writing, proofs for items 2 and 5 are already complete.

References

[Ede85] E. Eder. *Properties of substitutions and unifications.* J. Symbolic Comput. **1** 31–46, 1985.

[Kah15] W. Kahl. *Graph Transformation with Symbolic Attributes via Monadic Coalgebra Homomorphisms.* ECEASST **71** 5.1–5.17, 2015.

[Law63] F. W. Lawvere. *Functorial Semantics of Algebraic Theories.* Proc. Nat. Acad. Sci. USA **50** 869–872, 1963.

[Nor07] U. Norell. *Towards a Practical Programming Language Based on Dependent Type Theory.* PhD thesis, Dept. Comp. Sci. and Eng., Chalmers Univ. of Technology, 2007. See also http://wiki.portal.chalmers.se/agda/pmwiki.php.

[RS87] D. E. Rydeheard, J. G. Stell. *Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms.* In D. H. Pitt, A. Poigné, D. E. Rydeheard, eds., Category Theory and Computer Science, CTCS 1987, LNCS **283**, pp. 114–139. Springer, 1987.

[Szi83] J. Szigeti. *On limits and colimits in the Kleisli category.* Cahiers de Topologie ed Géométrie Différentielle **24**(4) 381–391, 1983.