Congruence Classes with Logic Variables

Joe Hurd*
Computer Laboratory
University of Cambridge
joe.hurd@cl.cam.ac.uk

12 July 2000

Abstract

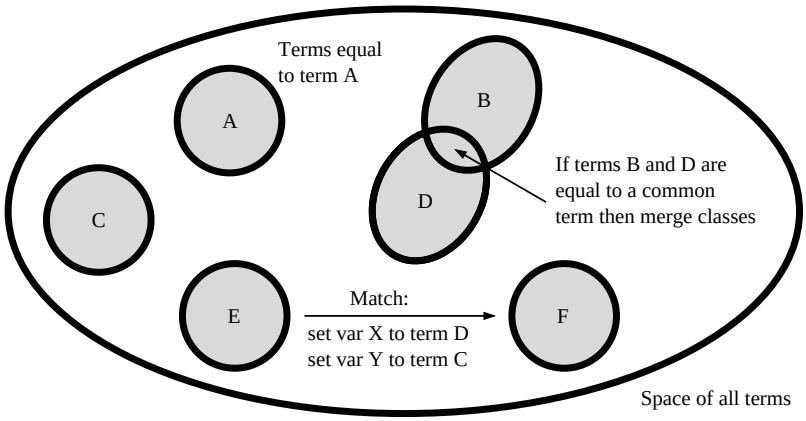
We are improving equality reasoning in automatic theorem-provers, and congruence classes provide an efficient storage mechanism for terms, as well as the congruence closure decision procedure. We describe the technical steps involved in integrating logic variables with congruence classes, and present an algorithm that can be proved to find all matches between classes (modulo certain equalities). An application of this algorithm makes possible a *percolation* algorithm for undirected rewriting in minimal space; this is described and an implementation in `hol98` is examined in some detail.

1 Summary

Blending together equality steps (Leibniz' rule of substituting equals for equals) and deduction steps (e.g., Modus Ponens or specialization) in a proof search is problematic. Equality tends to dramatically blow up the search space, because of the vast number of ways of expressing a given term.

In this work we take *congruence classes*—a way of storing terms that maximizes sharing and performs congruence closure—and show how they can be used when the terms contain logic variables. This makes them appropriate for storing terms in a deductive prover.

*Supported by an EPSRC studentship



Our results so far are:

- a matching algorithm between classes, guaranteed to find all matches (modulo certain equalities) between classes;
- a *percolation* algorithm that performs undirected rewriting on the classes (using the equalities represented by the classes);
- an implementation of the above as a derived rule in hol98, with results on some test cases.

The full paper is available at the following URL:

`http://www.cl.cam.ac.uk/users/jeh1004/research/papers/congruence1.html`