

## All-Termination(T)

Panagiotis Manolios, Aaron Turon (Northeastern University)

*This research appeared in TACAS 2009.*

ACL2, like most theorem proving systems, requires functions to be shown terminating before admission into the logic. Each proof of termination in ACL2 yields a *measured subset*, a subset of function parameters for which there is a measure showing the function terminating. Measured subsets play an important role in ACL2. They are used to guide rewriting heuristics (for example, to ensure their termination) and to derive and justify induction schemes.

Although ACL2 requires only one termination proof per admitted function, functions may have multiple measured subsets. For example, the following function has both  $\{i\}$  and  $\{list\}$  among its measured subsets:

```
(defun replace-ith (i v list)
  (declare (type (integer 0 *) i))
  (if (consp list)
      (if (zp i) (cons v (cdr list))
          (cons (car list) (replace-ith (1- i) v (cdr list))))
      nil))
```

Because of the importance of measured subsets when proving theorems about functions, it is to the user's advantage to provide as many measured subsets as possible. At present, each measured subset requires a separate witnessing measure, given through a separate proof of termination.

At the same time, sophisticated termination analyses are now available which are effective for proving termination in the vast majority of cases. For example, Manolios and Vroon's *calling context graph* (CCG) analysis proves termination automatically for about 98% of the ACL2 regression suite. Although such analyses are very convenient, they function as decision procedures, and thus it is not possible to extract nontrivial measured subsets from them.

These considerations led us to define the *All-Termination* problem: given a function (or nest of functions)  $f$ , enumerate all possible measured subsets of  $f$ . This generalizes the traditional decision problem for termination to a more useful enumeration problem. Moreover, we can consider the *All-Termination(T)* problem: for a termination analysis  $T$  and a function  $f$ , find as many measured subsets for  $f$  as possible, “using”  $T$ . Here we must say how to *restrict* the analysis  $T$  so that it justifies the termination of  $f$  using only a certain subset of  $f$ 's parameters. To solve All-Termination(T) efficiently, we seek ways to instrument  $T$  so that we need only execute it *once*, and afterwards can extract measured subsets from it.

We have carried out this research program for *size-change* (SCT) analysis, which underlies the CCG analysis already mentioned. We found that the complexity of All-Termination(SCT) is the *same* as the complexity of SCT itself, and gave an algorithm for it. We have implemented this algorithm as a new backend for CCG analysis, and executed it on the entire regression suite. Our results suggest that it is practical (taking only 30 seconds for the entire suite) and useful (finding nontrivial measured subsets for 90% of multiargument functions, and finding incomparable sets for 7% of them).