CARIBOO: A Multi-Strategy Termination Proof Tool Based on Induction

Olivier Fissore, Isabelle Gnaedig, Hélène Kirchner

CARIBOO :
A Multi-Strategy Termination Proof Tool
Based on Induction

1 A termination proof tool for rule-based programs

CARIBOO is a termination proof tool for rule-based programming languages, where a program is a rewrite system and query evaluation consists in rewriting a ground expression [3]. It applies to languages such as ASF+SDF, Maude, Cafe-OBJ, or ELAN.

By contrast with most of the existing tools, which prove in general termination of standard rewriting (rewriting without strategy) on the free term algebra, our proof tool, named CARIBOO (for **C**omputing **A**bst**R**action for **I**nduction **B**ased termination pr**OO**fs), allows proving termination under specific reduction strategies, which becomes of special interest when the computations diverge for standard rewriting. It deals in particular with :

- the innermost strategy, specially useful when the rule-based formalism expresses functional programs, and central in the evaluation process of ELAN,
- local strategies on operators, provided in OBJ-like languages, and allowing to control evaluation strategies in a very fine local way,
- the outermost strategy, useful to avoid evaluations known to be non terminating for the standard strategy, to make strategy computations shorter, and used for interpreters and compilers using call by name.

2 Proving termination by explicit induction

The proof technique of CARIBOO is backed by a generic inductive process, declinable for the different considered strategies [4, 2]. For proving that a term t terminates, we proceed by explicit induction on the termination property, on ground terms with a noetherian ordering $\succ$, assuming that for any t' such that $t \succ t'$, t' terminates (according to the given strategy). We first prove that a basic set of minimal elements for $\succ$ terminates. We require the subterm property for $\succ$, so the set of minimal elements is a subset of the set of constants of the signature. We then simulate the rewriting (according to the given strategy) derivation tree starting from a ground term t which is an instance of a term $g(x_1, \ldots, x_m)$, for all defined symbols g, and variables $x_1, \ldots, x_m$. Proving termination on ground terms amounts to prove that these rewriting (according to the given strategy) derivation trees have only finite branches.

Each derivation tree is simulated by a proof tree starting from $g(x_1, \ldots, x_m)$, and developed by alternatively using two main concepts, namely narrowing and abstraction, whose definition depends on the considered rewriting strategy. More precisely, narrowing schematizes all rewriting possibilities of the terms in the derivations. The abstraction process simulates the normalization of subterms in the derivations, according to the strategy. It consists in replacing these subterms by special variables, denoting any of their normal forms. This abstraction step is performed on subterms that can be assumed terminating by induction hypothesis. Our procedure terminates whith success when termination has been inductively established for each proof tree.

3 A constraint-based proof process

The induction ordering is not given a priori but is determined by ordering constraints set along the proof. Thanks to the power of induction, ordering constraints are often simpler than for other constraint-based termination proof rewrite methods, like the rule-orientation based method using simplification orderings: they are often satisfied by the subterm ordering or the embedding ordering, integrated in CARIBOO. Otherwise, they can be delegated to automatic external solvers like C*i*ME2 [1].

The narrowing process, well-known to easily diverge, is restricted for the innermost and local strategies, by abstraction constraints, expressing which terms abstraction variables represent the normal form of. Abstraction constraints are used only for proofs whose narrowing process diverges.

Proofs involving only ordering constraints satisfied by integrated orderings or by C*i*ME2, are completely automatic.

4 A flexible technique allowing integrating other approaches

The proof method of CARIBOO allows the combination with auxiliary termination proofs with a different technique: when the induction hypothesis cannot be applied on a term u during the proof, it is often possible to prove termination (according to the given strategy) of any ground instance of u by any free other way, like testing its ground reducibility or studying rules involved in its reduction.

This flexibility of our method, allowing to integrate other termination proof approaches in the main inductive proof process in a natural way, gives us a formal framework for different termination tools to efficiently cooperate, unlike other methods that, when failing, require the user to find another proof technique by himself.

5 A user-friendly ELAN reflexive tool

The proof tool consists of two main parts :

- the proof procedure, written in ELAN, generating the proof trees and writing information (constraints and proof trees) both on the standard output and input files : our abstraction and narrowing-based proof mechanism is formalized with inference rules, backed by the same principle for the different studied rewriting strategies, and a strategy to apply them, dependent on the rewriting strategy. As ELAN was precisely designed for prototyping rule and strategy-based algorithms, the proof procedure is directly encoded through ELAN rules and an ELAN strategy to apply them.

- a graphical user interface (GUI), written in Java, making it possible for the user to follow each step of the proof process : which defined symbols have already been handled and, for each of them, the proof tree together with the detail of each state.

CARIBOO is reflexive in the sense that its rule-based proof process can be applied to prove termination of ELAN programs.

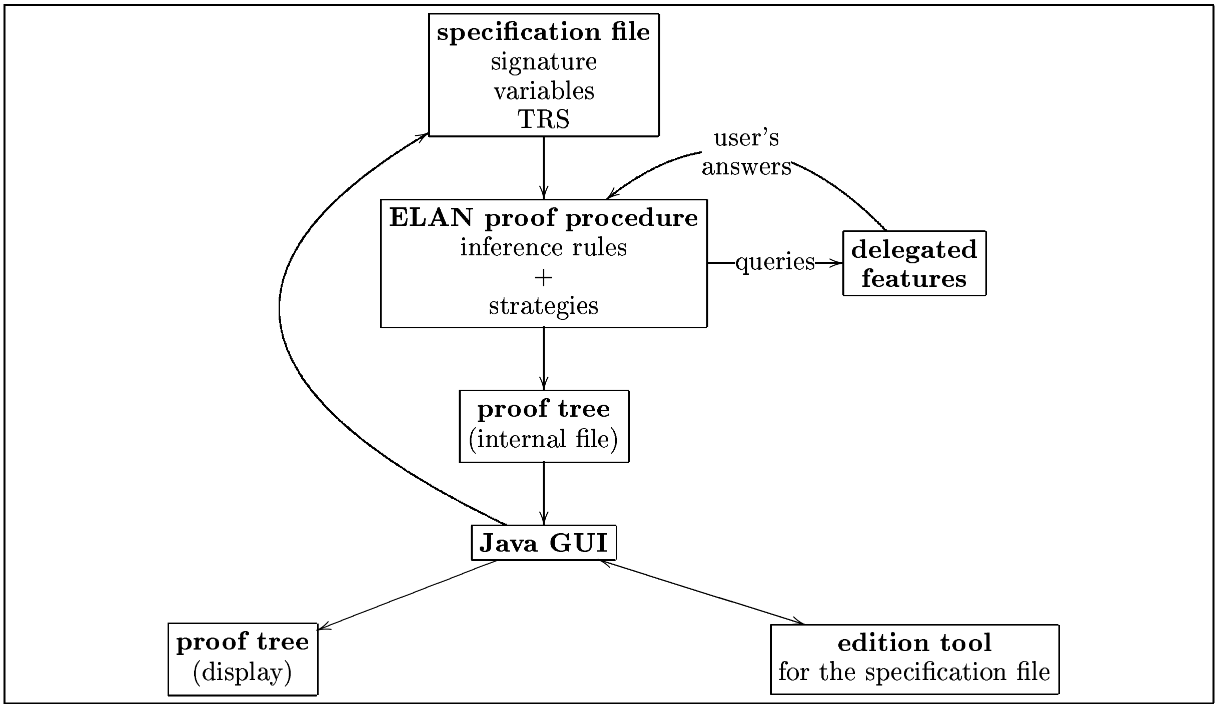
6 How CARIBOO works

The ELAN proof procedure expects the user to give the TRS which he wants to prove termination of, i.e. the rewrite rules of the system along with the variables and the signature of the algebra.
An edition tool has been written in Java to help the user enter these data, which are then transformed into the ELAN specification used by the main program.

During the process, the ELAN proof procedure builds the proof tree, and writes the detail of each state into a file. This information is then picked up by the Java GUI that displays in a graphical way the evolution of the proof process.

To handle the delegated features, i.e. to prove termination of a term by an external criterion, or for abstraction constraint solving, the proof procedure directly communicates with the user through the window in which it has been launched.

Then, once the proof process is achieved, the proof tree has been generated by the ELAN proof procedure for each defined symbol and displayed by the Java GUI. We can get the detail of each state of these trees. If none of the proof trees contains a failure state, then the TRS given as input is terminating w.r.t. the given strategy. The trace can be saved in text or LaTeX format.



CARIBOO will also soon offer a termination proof procedure for full ELAN strategies, transforming an ELAN strategy into a simpler one, and applying sufficient conditions for termination on the simplified form.

References

[1] Benjamin Monate et Xavier Urbain Evelyne Contejean, Claude Marché. Cime version 2, 2000. Version préliminaire disponible à http://cime.lri.fr/.

[2] O. Fissore, I. Gnaedig, and H. Kirchner. Termination of rewriting with local strategies. In M. P. Bonacina and B. Gramlich, editors, *Selected papers of the 4th International Workshop on Strategies in Automated Deduction*, volume 58 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2001.

[3] O. Fissore, I. Gnaedig, and H. Kirchner. CARIBOO : An induction based proof tool for termination with strategies. In *Proceedings of the Fourth International Conference on Principles and Practice of Declarative Programming (PPDP)*, Pittsburgh, USA, October 2002. ACM Press.

[4] O. Fissore, I. Gnaedig, and H. Kirchner. Outermost ground termination. In *Proceedings of the Fourth International Workshop on Rewriting Logic and Its Applications*, Pisa, Italy, September 2002. Electronic Notes in Computer Science. To appear.