TERSE : A Visual Environment for Supporting Analysis, Verification and Transformation of Term Rewriting Systems

Nobuo KAWAGUCHI   Toshiki SAKABE   Yasuyoshi INAGAKI

Department of Information Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01 JAPAN
E-mail: kawaguti@nuie.nagoya-u.ac.jp

1 Introduction

Term rewriting systems(TRS)[1] can be widely applicable in many areas of computer science such as equational logics, theorem proving, algebraic specification, program verification, transformation and synthesis, etc. In such applications, an environment for term rewriting with user friendly graphical interface is strongly required for analyzing structure of terms and rewriting processes. Most TRS implementations developed so far[3, 4] are text-based ones, and hence they do not provide the sufficient supports for analyzing structure of terms nor rewriting sequences.

We have developed a visual environment for term rewriting computation. The environment (TERSE:TErm Rewriting Support Environment) provides various kinds of visualization to support analysis, verification and transformation of TRS. Currently, five sorts of visual viewers are implemented in TERSE. Several kinds of automated termination proof and transformation algorithms are also included in the environment. TERSE is implemented with CML(Concurrent ML[5]) with eXene[6] library. Using higher order features in CML makes it easy to modify and extend the environment.

2 Overview of TERSE

TERSE is composed of four parts, Core part and three Studios(Figure 1). Core part have the term rewriting engine and Main Window. Main Window can handle multiple terms, rewriting rules and rewriting strategies. The user can perform text-base term rewriting in Main Window and enter to any Studios for further analysis.

2.1 Visualization Studio

Visualization Studio is currently composed of four viewers(Term, Sequence, Relation, Statistics). Fig 2 shows Term Viewer. The structure of a term is displayed by a tree with information such as redex occurrence, sort of function symbols and subterm abbreviation. We call this special drawing method for terms as

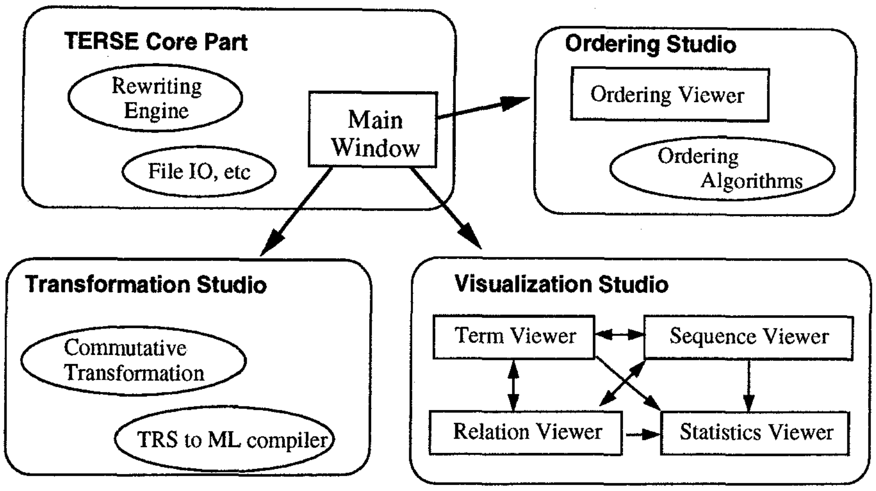

Fig. 1. Modules of TERSE

Term Visualization. One can call other three viewers from Term Viewer with the current displayed term.

Sequence Viewer shows the rewrite sequences of terms with Term Visualization. Visual analysis on rewrite sequence provides the insight of effective transformation.

Relation Viewer displays the graph of rewriting relations. Each node on the graph represents one term and the arcs represent the rewriting relation. The root node is starting point of rewriting. Visualization of rewriting relations can provide various information for evaluating rewriting strategies.

Statistics Viewer shows statistic information of the term by line-chart. Each line on the chart shows the size, length, width and number of redexes of the term. The functions for acquising for these statistic values can be exchanged with any user configured function.

2.2 Ordering Studio

Ordering Studio implements various kinds of termination algorithms[2]. It also has a Ordering Viewer for visualizing the recursive structure of path orderings.(Shown in Fig.3) Two terms are displayed at the same window and overlapped colored rectangles describe the comparison of subterms. This visualization helps to determine the precedence of function symbols and to analyze the ordering.

2.3 Transformation Studio

Transformation Studio has two transformation methods. Commutative transformation is an automated transformation algorithm for TRS optimization. TRS

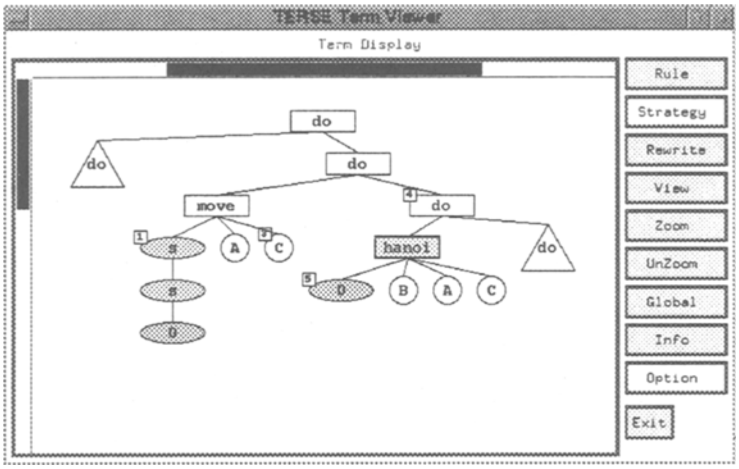

Fig. 2. Term Viewer

to ML compiler converts TRS rules into ML programs. The resulting program can be evaluated under the environment.

3 Implementation

TERSE is implemented on CML(Concurrent ML)[5] with eXene[6] Library. CML is a concurrent functional language and eXene is an X-window library built on CML. Implementation of TERSE is clearly modularized by MVC(Model, View, Controller)model. Powerful datatype facility makes ML as one of the most powerful and easy prototyping languages for symbolic computation(Model). Combination of CML and eXene makes GUI(View and Controller) programming on ML practically. The code size of TERSE is about 8000 lines

Currently, many (text-based) symbolic computation systems are well implemented in ML. Visualizing methods and GUIs provided in TERSE might be also applicable for those systems.

4 Concluding Remarks

We have proposed TERSE, a visual environment with GUI that supports design and analysis of TRS. TERSE provides various operations on visualized terms and TRSs such as customizing visualization style, controlling rewriting steps, analyzing rewriting processes, and so forth.

The implementation is modularized clearly by MVC model for the convenience of further refinements and reuses.

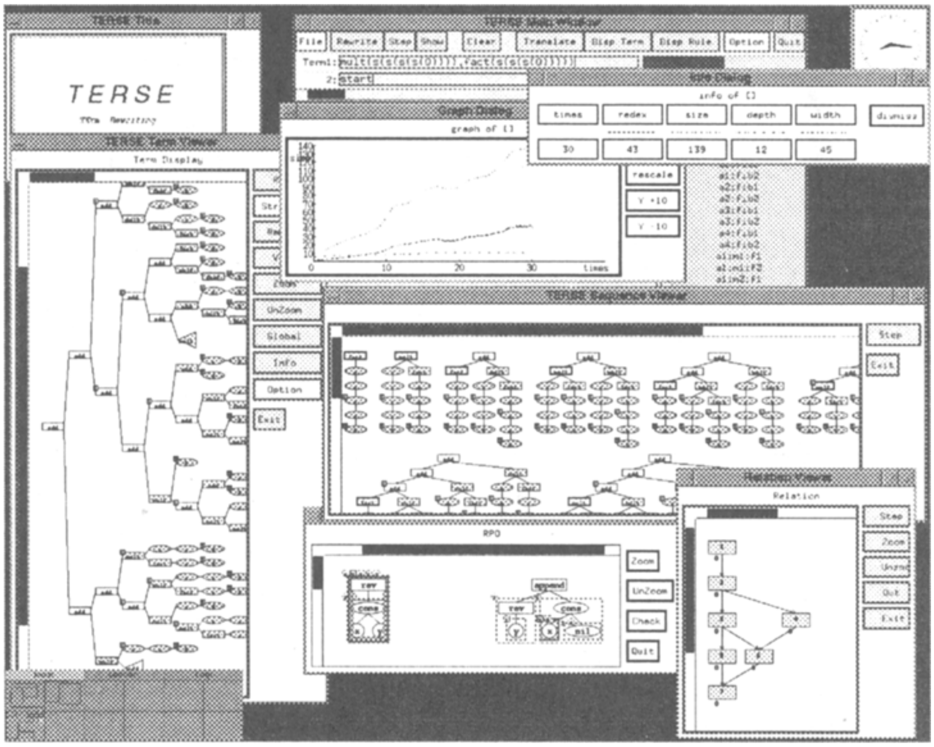

Fig. 3. A screen shot of TERSE

References

1. Huet, G. , "Confluent Reductions:Abstract Properties and Applications to Term Rewriting Systems", J.ACM, Vol.27, No.4, pp797-821 (1980).
2. Dershowitz, N. , "Termination of Rewriting" , J. Symbolic Computation, Vol.3, pp69-116 (1989).
3. Matthews,B. : MERILL: An Equational Reasoning System in Standard ML, *Rewriting Techniques and Applications,* Lecture Notes in Computer Science, No. 690, pp. 441–445(1993).
4. Bundgen,R. ,Reduce the Redex → ReDuX , Rewriting Techniques and Applications, Lecture Notes in Computer Science, No. 690, pp. 446–450(1993).
5. Reppy,J.H., "CML: A higer-order concurrent language", In Proceedings of the ACM SIGPLAN'91 on PLDI, pp293-305 (1991).
6. Reppy,J.H., Gansner,E.R., "The eXene Library Manual(Version 0.4)", AT&T Bell Laboratories (1993).
7. Kawaguchi,N., Sakabe,T. and Inagaki,Y. ,"TERSE: TErm Rewriting Support Environment" ,Proceedings of the 1994 ACM SIGPLAN Workshop on Standard ML and its Applications, pp91-100 (1994).
8. Kawaguchi,N., Sakabe,T. and Inagaki,Y. ,"Visual Support Methods for Analysis, Verification and Transformation of Term Rewriting Systems" ,JSSST Computer Software, Vol.13, No. 1, pp.23-36 (1996).(In Japanese)