

The Semantically Guided Linear Deduction System

Geoff Sutcliffe

Department of Computer Science, The University of Western Australia
Nedlands, Western Australia, 6009. Email : geoff@cs.uwa.oz.au

1. Introduction

The Semantically Guided Linear Deduction System (SGLD) is the successor of GCTP [Sutcliffe, 1990]. SGLD has been constructed by imposing semantic guidance onto a chain format linear deduction system called Guided Linear Deduction (GLD). GLD is broadly based on the Graph Construction (GC) procedure [Shostak, 1976], but has added features which improve performance.

2. Deduction Operations

GLD's eight deduction operations consist of six inference operations - extension, unit subsumed extension, A-reduction, identical A-reduction, C-reduction, and identical C-reduction; and two bookkeeping operations - A-truncation and C-truncation. The extension operations are, in combination, equivalent to the extension operations of other chain format systems. GLD's reduction operations combine features from Model Elimination (ME) [Loveland, 1969] and the GC procedure. A significant feature of GLD's non-compulsory reduction operations is the use of a selection rule, which provides a search guidance point not available in most other chain format systems. The reduction operations work in tandem with A-truncation to implement re-use of deduced information.

Orthogonal to the operational divisions, the deduction operations may be split into two groups based on whether or not alternative successor center chains need be considered once the operation has been completed. The latter group are the *compulsory* operations, being unit subsumed extension, identical A-reduction and identical C-reduction. An important difference between the non-compulsory and the compulsory inference operations is that the non-compulsory inference operations operate on a selected B-literal in the rightmost cell of a center chain, while the compulsory inference operations may eventually use any B-literal in a center chain.

3. Deduction Chunks

The desirable feature of using coarse grain deduction steps in a deduction system has been approached in GLD by combining multiple deduction operations into indivisible deduction chunks. The philosophy underlying GLD's operation chunking is that no center chain is stored whilst it contains a literal that can be removed by a compulsory operation. After each non-compulsory operation, a maximal sequence of compulsory operations is performed before the resulting center chain is stored. The intermediate chains deduced whilst building the maximal sequence are discarded. The initial non-compulsory operation and the sequence of compulsory operations form a deduction chunk.

4. Search Strategy

For each chunk of a GLD deduction, two choices have to be made. The first is to select a B-literal for the non-compulsory operation; the second is to choose an order in which alternative successor center chains are considered. GLD uses two methods for selecting a B-literal and two methods for ordering alternative successors. These in combination provide four possible search styles - literal-selected, literal-ordered, cell-selected and cell-ordered.

The literal-selected and literal-ordered search styles use a trivial B-literal selection method, simply taking the rightmost B-literal. This provides no search guidance. In the cell-selected and cell-ordered styles, the B-literal selection aims for the selection that is most likely to lead to failure. For each possible selection of B-literal the set of successor center chains is deduced, the heuristic value of each successor in each set calculated, and the best value in each set noted as the heuristic value of the set. The B-literal whose successor set has the worst value, is selected.

The literal-selected and cell-selected search styles use a default ordering of alternative successor center chains. The default ordering is guided by (i) a 'fewest-literals preference' maxim, and (ii) by avoiding changing scope values (see below) if possible. The literal-ordered and cell-ordered search styles order the alternative successor center chains in order of worsening heuristic value.

The overall search strategy of GLD is a modified consecutively bounded depth first search which places a bound on the number of center chain A- and B-literals.

5. Re-use of Deduced Information

GLD uses a combination of ME's lemma mechanism and the GC procedure's C-literal mechanism, to re-use deduced information. Because of the advantages of the lemma mechanism over the C-literal mechanism (principally, lemmas are persistent), preference is given to the creation of lemmas. The addition of lemma chains to the input set is limited by the use of forward subsumption. Further, non-unit lemmas are only added to the input set if they backward subsume at least one existing input chain. If no existing input chains are backward subsumed then a C-literal is inserted into the center chain. In this manner the number of non-unit input chains does not increase. Unit lemmas are always added to the input set if they are not forward subsumed. To implement the combined mechanism, A-literals maintain a scope value. The C-point of an A-truncation is determined from the scope values.

6. Admissibility Restrictions

The admissibility restrictions in GLD are based on those of the GC procedure. The GC procedure specifies that no two non-B-literals in any center chain may have identical atoms. GLD imposes these and five new restrictions. (i) No A-literal may be to the left of an identical B-literal. (ii) No C-literal may be to the left of an identical B-literal. (iii) No

B-literal may be in the same cell as a complementarily identical B-literal. (iv) No B-literal may be in the cell immediately to the left of a complementarily identical A-literal. (v) No A-literal formed in a non-unit extension may be complementarily subsumed by a unit input chain.

7. Semantic Guidance

SGLD is GLD using two forms of semantic guidance : (i) Truth value deletion in some parts of deductions, and (ii) A heuristic function that has a preference for FALSE chains.

7.1 Linear Input Analysis

There are syntactically identifiable situations in which A- and C-reduction does not occur in GLD, i.e. situations in which linear-input subdeductions are performed. Three methods of analysing sets of input chains have been developed for detecting these situations. The first method (Horn subset analysis) focuses on Horn input chains while the second (LISS analysis) and third (LISL analysis) are successive generalisations of the first method. The start of a linear-input subdeduction is identified by the selection of a linear-input B-literal for an extension operation. This literal is called the top literal of the subdeduction. A linear-input subdeduction ends when its top literal is truncated.

In a linear-input subdeduction the reduction operations can be explicitly ignored, so that no effort is expended attempting to find A-literals or C-literals to reduce against. If Horn subset analysis is used then only the positive literals of Horn subset input chains need ever be considered when searching for a suitable input chain in an extension operation. A more significant benefit that may be derived from linear-input analysis is the completeness of a truth value deletion strategy in linear-input subdeductions.

Def'n : The *rightwards subchain* of a literal in a center chain consists of the literal and all literals to its right.

A truth value deletion system which requires all rightwards subchains of the top literal in a linear-input subdeduction to be interpreted as FALSE, in all models of the side parent chains used in the subdeduction, is complete. Such a system is integrated into SGLD's search guidance system : the heuristic function fails to return a value if a center chain has no ground instances which are acceptable to the deletion system, and the center chain under consideration is rejected. Linear-input analysis and the truth value deletion system are described in [Sutcliffe, 1992].

7.2 FALSE Preference

Truth value deletion rigidly expects chains to be interpreted as FALSE in an interpretation. By softening this expectation to having a preference for parent chains that are interpreted as FALSE, a new strain of truth value guidance systems is formed. These guidance systems are called *truth value preference systems*. A truth value preference system is used in SGLD by using a heuristic function which calculates the FALSity level of center chains. The FALSity level of a center chain is a non-worsening function of the FALSity levels of its ground universe instances; the FALSity level of a ground universe instance of a center chain is a non-worsening function of the FALSity levels of its literals;

and the FALSity level of a ground literal is FalseScore if the literal is interpreted as FALSE, TrueScore if the literal is interpreted as TRUE. The values of FalseScore and TrueScore are parameters to the preference system. FalseScore is a better FALSity level than TrueScore. The effect of the FALSE preference is to minimise the number of reduction operations in a deduction.

8. Conclusion

SGLD has been implemented in Prolog, and performance testing shows the efficacy of using semantic information to guide search. As well the features described above, SGLD has facilities for imposing sort value deletion and for embedding equality. Sort value deletion is imposed via the same mechanism as the truth value deletion, but throughout deductions. Equality is embedded via an extension of the RUE and NRF inference rules [Digricoli, 1979].

References

- Digricoli V.J., Automatic Deduction and Equality, In Martin A.L. (Ed.), *Proceedings of the Annual Conference of the ACM* (Detroit, MI, 1979), ACM Press, New York, NY, (1979), 240-250.
- Loveland D.W., A Simplified Format for the Model Elimination Theorem-Proving Procedure, In *Journal of the ACM* 16(3), ACM Press, New York, NY, (1969), 349-363.
- Shostak R.E., Refutation Graphs, In *Artificial Intelligence* 7, Elsevier Science, Amsterdam, The Netherlands, (1976), 51-64.
- Sutcliffe G., A General Clause Theorem Prover, In Stickel M.E. (Ed.), *Proceedings of the 10th International Conference on Automated Deduction* (Kaiserslautern, Germany, 1990), (Siekmann J.H. (Ed.), *Lecture Notes in Artificial Intelligence* 449), Springer-Verlag, New York, NY, (1990), 675-676.
- Sutcliffe G., Linear-Input Subset Analysis, In Kapur D. (Ed.), *Proceedings of the 11th International Conference on Automated Deduction* (Saratoga Springs, NY, 1992), Springer-Verlag, New York, NY, (1992).