

FENDER: An approach to theory restructuring (extended abstract)*

Edgar Sommer

GMD — German National Research Center for Computer Science, AI Division
(I3.KI), Schloss Birlinghoven, 53757 St. Augustin, Germany, email sommer@gmd.de

Theory Restructuring is an emerging research issue whose task is to transform a given knowledge base without changing coverage of a goal concept or concepts, i.e. to in some sense improve the theory's structure, without changing its inferential outcome [Sommer 94a]. In [Sommer 94b], the notion of stratification was introduced, essentially separating the inductive properties of Inverse Resolution [Muggleton/Buntine 88] from its (re-)structuring properties. Following a specific strategy, FENDER performs a number of inter- and intraconstruction steps on a given theory, restructuring by introducing new intermediate concepts, *retaining* the set of computed answers of the theory. In other words, FENDER performs inverse resolution on given clauses, without generalizing them. The result is a new inferential structure that is deeper and more modular, and possibly easier to understand and maintain. The new intermediate concepts are intensionally defined and put to immediate use; they make implied relationships explicit by exploiting similarities and differences between original clauses of the theory.

Stratification with FENDER The basic task description: take the rules of a KB; aggregate some of the premises to form new concepts; rewrite the ruleset using these concepts. As will be seen, the decisions that need to be made along the way will be guided by two intuitions: (1) new concepts that find heavy use in the KB are to be preferred, and (2) new concepts that allow the suppression of variables in the original rules are to be preferred. The first is quite straightforward and akin to the idea of compactness in DUCe and CIGOL. The second may seem out of the blue, but most strongly characterizes the approach taken here. Consider that there is natural way of classifying the variables occurring in a rule: head and non-head. Head variables are those that appear in the conclusion of the rule, non-head those occurring *only* in the premise. If we take a goal-directed view of a KB, non-head variables are of "lesser" interest and complicate the rules they appear in — they represent objects in the domain needed for certain inferences, but not needed in the final answers. The approach taken here uses this distinction to solve some basic identification problems (which variable in one rule does variable X in another rule correspond to?) and — more significantly — to decide among alternative common partial premises (CPPs), the building blocks for new intermediate concepts.

* Full length version available via <http://nathan.gmd.de/persons/edgar.sommer.html>.
Less up-to-date but more detailed description of FENDER published as [Sommer 94b].

A **basic stratification procedure** is detailed in Fig. 1. Stratification performed in this way groups some of the conditions of a rule (step 2), interprets these as concepts (step 4a), and rewrites all affected rules in terms of these new concepts (step 4b). When the process is repeated, some of the new concepts may be combined (with others or with original premise literals) to form further concepts, yielding a deeper inferential structure. The new concepts make recurring conditions in the original rules explicit.

1. collect the set of rules disjunctively defining a concept
(all subsequent operations are performed on a copy;
only the last step (7) actually changes the knowledge base)
2. **[CPP collection]** for each rule
 - (a) collect the rule variables
 - (b) for each rule variable
 - i. collect the premise literals it appears in
 - ii. if this is a new partial premise
 - A. search the other rules for occurrence of this partial premise
(rules affected by this partial premise)
 - B. remember partial premise and affected rules
3. **[CPP grouping and selection]** select desired CPPs (see next sections)
4. **[preliminary translation]** for each partial premise
 - (a) construct a new concept name and define new concept
 - (b) in each affected rule, replace partial premise with new concept
5. repeat (from 2) until no new partial premises are found
6. **[final modifications]** (see next sections)
7. **[actual knowledge base translation]**

Fig. 1. *A basic stratification procedure*

Decisions to be made A number of problems and pragmatic issues must be taken into consideration. These are generally valid in a folding task. Specific solutions to them as implemented in FENDER are given in the next section; they appear in the basic procedure (Fig. 1) as steps 3 and 6.

- CPPs collected in this straightforward manner may “overlap”: they share some premise literals. Only one in a group of such overlapping CPPs can be applied to the affected rules.
- A CPP in some way defines an intermediate concept, a (possibly) useful relationship between variables of the original rules. Which of the many possibilities are to be preferred?
- Does the new concept necessarily concern *all* of the variables occurring in the CPP? Not all of the variables may be of interest in the next inference step, so they could be dropped from the new concept.
- The procedure does not construct disjunctive intermediate concepts. This means that the original set of goal-concept rules is transformed one-to-one into rules with shorter premises. This does not account for rules (or CPPs) that differ “only slightly”. Such slight differences can be noted in the form of disjunctive definitions, yielding more meaningful concepts, and reducing the number of top-level rules.

Strategic decisions made Specific decisions made in FENDER concern steps 3 and 6 in Fig. 1. First, following the motivation at the beginning, CPPs collected around non-head variables are preferred (compare step 2). The iteration process ends when no such **non-head CPP** is found. This is motivated by the hope of being able to suppress some such variables in the rules of the KB. It also solves half of the overlap problem mentioned above. The other half is solved by preferring the most popular CPP in an overlap group, i.e. the one which occurs most often in the original rules². Grouping itself is done in simple manner: all CPPs that share one or more literals form a group (of which only one can be applied to the affected rules). Of course, much more sophisticated methods — subsumption tests, for example — are conceivable. But this is efficient and guaranteed to find the most popular, non-overlapping, non-head CPP(s) in each pass over the rule set (iterations of loop steps 2 to 5 in Fig. 1).

Second, once a non-head CPP has been decided upon in this manner, each affected rule is inspected with the following question in mind: does the seed variable occur anywhere in the premise, other than in that part represented by the CPP? If not, it can be “suppressed”: the new intermediate concept to be defined by the CPP will not include the seed variable in its head. The original rules, rewritten using this new concept, will be simpler (and easier to understand & modify) because they concern less variables.

Third, the individual CPPs of an overlap group are compared. All those that differ in one conjunct only are combined to form the definition of a *disjunctive* intermediate concept. This can result in further enrichment of the inferential structure (fanning-out as opposed to depth). When stratification finds no new CPPs, all rules are compared a final time (step 6). Those that differ only in a constant are reformulated in the following manner: the constants are collected to define a new unary concept; using this concept in the definitions of the affected rules has the effect that they dissolve into one, thus further simplifying and modularizing the theory.

Conclusion & related and future work The three transformation operators FENDER employs can be cast in the general framework of inverse resolution. The main operation, replacing a CPP in several rules with a new concept, can be seen as a form of *interconstruction*, which is mentioned as a complement to *intraconstruction* in [Muggleton/Buntine 88], but not implemented. The second, using several similar CPPs in an overlap group to form the definition of a single *disjunctive* intermediate concept, can be seen as a form of *intraconstruction*, applied to a new concept rather than the goal. The third, combining all rules that differ only in a constant — by constructing a new unary concept that covers these constants — is similar in scope and effect to the form of *intraconstruction* implemented in CIGOL. The unit clause assumption made in CIGOL has the effect that intraconstruction is applied only to clauses made up of exactly the same predicate symbols, so that its (re-) structuring capabilities are quite limited in

² An interactive version of FENDER lets the user select one of the less popular CPPs in an overlap group if deemed more valuable.

practice. Furthermore, the new concepts it finds are defined by unit clauses — i.e. by instance, while FENDER's definitions are rules.

One of the main characteristics of FENDER as a whole is that more effort is put into finding CPPs; in CIGOL and LFP2 [Wirth 89], the common generalization of the given clauses is taken. If none common to *all* clauses exists, then no restructuring is possible. In FENDER, CPPs are constructed around the individual variables of the given clauses, which offers the possibility of more meaningful intermediate concepts, of using CPPs common to some rather than all clauses, of adding more than one layer of inference, and of suppressing non-head variables.

As a first approach to theory restructuring, Fig. 1 sets up a framework for experimentation with different strategies. Problems that must be addressed by them are identified, and the specific strategy implemented in FENDER, based on CPP popularity and the elimination of non-head variables, shows some promising results. FENDER reduced the size of, and introduced useful concepts and new structure into, a KB governing access to telecom switches [Sommer et al. 94]. FOIL5 was not able to learn good rules in the original representation of the telecom KB, but found perfect definitions using the new concepts.

In a step towards offering a knowledge base restructuring tool chest, a complementary *unfolding* procedure has been incorporated in MOBAL [Morik et al. 93] along with FENDER. It is able to re-derive the original KB from the one produced by FENDER. To round off the picture, MOBAL also offers operators that analyze and restructure with respect to different notions of redundancy [Sommer 94a]. A set of criteria that attempt to quantify the quality of a KB's structure with emphasis on understandability has been implemented in MOBAL [Sommer 95]. This will aid the comparison of different KB forms produced by restructuring operators. For instance, if *number of premise literals*, *number of (non-head) variables*, *number of goal-concept rules* and *theory size* are valid measures of quality, then FENDER can be shown to improve these criteria and hence theory structure.

References

- [Morik et al. 93] K. Morik, S. Wrobel, Jörg-Uwe Kietz, and W. Emde. *Knowledge Acquisition and Machine Learning*. Academic Press, London, 1993.
- [Muggleton/Buntine 88] Stephen Muggleton and Wray Buntine. Machine Invention of First-Order Predicates by Inverting Resolution. In *Proc. Fifth Intern. Conf. on Machine Learning*, San Mateo, CA, 1988. Morgan Kaufman.
- [Sommer et al. 94] E. Sommer, K. Morik, J.M. Andre, and M. Uszynski. What On-line Learning Can Do for Knowledge Acquisition. *Knowledge Acquisition*, 6:435–460, 1994.
- [Sommer 94a] E. Sommer. Restructuring in Horn clause knowledge bases. Technical report, ESPRIT Project ILP (6020), 1994. ILP Deliverable GMD 2.1.
- [Sommer 94b] E. Sommer. Rulebase Stratification: an Approach to theory restructuring. In *Proc. 4th Intl. Workshop on Inductive Logic Programming (ILP-94)*, 1994.
- [Sommer 95] E. Sommer. An Approach to Quantifying the Quality of Induced Theories. Technical report, GMD, 1995. (forthcoming).
- [Wirth 89] Ruediger Wirth. Completing Logic Programs by Inverse Resolution. In Katharina Morik (ed.), *Proc. Fourth European Working Session on Learning (EWSL-89)*, pp. 239–250, London/San Mateo, CA, 1989. Pitman/Morgan Kaufmann.