Review of "Implementing Classical Constructive Negation"

by Susana Munoz-Hernandez and Juan José Moreno-Navarro

OVERALL EVALUATION: REJECT

Summary

This paper addresses the important, and somewhat forgotten, topic of constructive negation. As the authors clearly identify, there have been several proposals in the literature to extend Prolog and other logic programming languages with a proper implementation of negation as failure/default negation. Unfortunately, these have not been implemented or surpassed simple prototypes.

The authors describe a complete Prolog implementation of constructive negation, however the presentation is somewhat cumbersome/confusing which lead to rejection of the paper. In particular, sections 2.2 and 2.3 require substantial improvements. There is a mixture of theoretical definitions with implementation which sometimes do not match, besides some missing definitions.

After careful revision and extension this can be a very valuable work and implementation for logic programming practicioners.

The paper is not at all self contained and details are lacking, which require substantial effort from the reader. No proof of correctness of the algorithm is attempted (not even informally). The examples are also not illustrative enough of all the implementation/theoretical underlying issues; most of the times they are trivial and the difficult details are not addressed.

Detailed Comments

The introduction is quite good and provides an overview of the literature in the topic which is rather complete and accurate. However, the level of detail is probably too much for an introduction and it would be better to have a section of comparisons after the proposed method of the authors. This would provide the reader with additional background which would let him better appreciate the comments and material of this section.

Definitely, it is lacking a section where the authors explain Chan's method and techniques. I do not agree with the authors comment that this will make the paper too big. In fact, it is essential since these are techniques with almost 20 years and are extensively used in the paper. Furthermore, it is also missing more background on solving constrains in the Herbrand Universe (finite trees) with infinite number of functors which is required to understand some of the simplifications/methods adopted by the authors, and in particular the computational complexity of the satisfiability problem in this domain (NP-complete). Moreover, the use of Rational Trees instead of Finite Trees is not discussed, but it is pretty relevant since Prolog unification does not perform occurs check.

Regarding Section 2, there are several problems that require attention. First, the authors never define explicitly the syntax they are using. Apparently, it is a superset of Prolog where, for instance, existential quantifiers are sometimes explicitly stated. For instance, in page 11, the predicate frontier is returning an existentially quantified goal. However, neither the definition of frontier introduces them nor the corresponding implementation requires them. Furthermore, the syntactic form of disequations is only introduced in page 15, and it is not explained why the authors adopted this form.

Regarding the definition of frontier, there are some typos (see next section) but the authors should mention that the implementation is exponential, because of the application of de Morgan's law. Again, are existential quantifiers introduced in the frontier of the goal for the variables which occur in the body and not in the head of rules? This is particularly important since most of the problems in the implementation of constructive negation is the handling of these variables. Apparently, the authors sometimes assume that these quantifiers are explicit (e.g. in page 12) and sometimes not (e.g. page 13 and in the implementation). This is very confusing to the reader.

Page 12 and 13, require substantial rewriting, namely the classification of variables and how they are obtained. This is not clear at all from the existing text (see comments in the next section). The authors also present some elimination of variables, equalities and disequalities. The soundness of the method should be proved, or at least put a reference to the literature, in particular for the disequalities (an example would help...).

In Section 2.3, the authors present what they call the classical constructive negation algorithm: please put a reference. The formula is again reorganized, and it is not explained why this is not done before. It should also be ex-

plained the meaning of "with variable in $ExpVars$": at least one variable? All variables? Subsequently, the constructive negation of the divided formula is presented without further explanation (I believe it relies on a result by Chan). Notice that the formulas at the top of page 14 have an existential quantifier which strangely disappears/moves inside C_i.

At beginning of page 15, the authors present the constraints, and in fact the answer is the union of several disequations. This was not introduced before. Afterwards, in the negation of $\bar{D}_{imp}$ disequalities also allow an existential quantifier after the universal one, which is not compatible with the previous form of constraints. This should be explained and why it is necessary to consider such cases. In fact, the variables Z_i are not explained where they occur, as well any restrictions that they must obey. Therefore, the comment at the end of this item is not clear at all. This should be based on a formal result, which should be either cited or shown. As a side remark, is there any preferred ordering of generation of the solutions obtained from the negation of $\bar{D}_{imp}$? For instance, one might prefer solutions with less universially quantified variables.

The negation of $\bar{D}_{exp} \wedge \bar{R}_{exp}$ requires further explanation, namely how the answers are constructed from the formula (the authors never present a formal notion/syntax of answers that can be generated from their procedure). As the authors state, this is the place where previous implementations have failed. So, please explain this very carefully with all the detail. I was not able to understand how is this done, and how the universal quantifier is applied to the answers obtained from the recursive call to the constructive negation procedure. Please put illustrative examples.

Section 3.2 should be extended with the details about the implementation of handling quantified disequations. In particular, the form of constraints is not compatible with the disequations $\forall \bar{W}_i \exists \bar{Z}_i Y_i \neq s_i$ which appear before in Section 2.3. Also, the unification algorithm should also be explained since the goal $X = / = s(fA(Y)), Y = s(foo)$ should fail. The authors also assume that the reader is acquainted with attributed variables (probably most of the readers are not).

Regarding sections 4 and 5, I do not have any extensive comments except that the examples are probably too simple in order to illustrate the full potential of the method, and also for benchmarking. Section 5.1 requires some background of the reader which could be explained in one or two sentences, e.g. what is the upper cost analysis? What does the predicate `findnsols/4` do?

The appendix requires some careful revision of the english. Furthermore, in example "Even and Odd" the answers contain new notation, e.g. $X/0$ and ${}_A/s(0)$. What does that mean?

Typos and other comments

Page 2, line 2: saying that declarative programming is similar to human reasoning is probably too daring.

Page 2, last paragraph: You should clarify what do you mean by "restriction." I think the beginning of this paragraph is very confusing, when the authors talk about "theoretical point of view" and "semantic level". What do you understand by "theoretical point of view"? Isn't the semantic level a major part of the "theoretical point of view" ? In fact, it is the theory that helps explaining the difficulties.

Page 3: Question mark missing at the end of the first sentence.

Page 3, paragraph 2: the term "inequality" is used instead of "disequality" adopted in the rest of the paper.

Page 5: The name "Maluszynski" is not spelled correctly.

Page 10: In Definition 2 is $\bar{X}$ a sequence of complex terms or only variables? What is $\bar{X}^i$? I also believe that the bodies of the rules should be $C'_1, \ldots, C'_N$ instead of $C_1 \ldots C_N$. You should also put the equalities before C'_i, according to your convention afterwards.

Page 12, 3rd paragraph: "organise" instead of "organice".

Page 12: In implementation details, $GoalVars$ is not defined before. Furthermore, the authors now adopt a different convention for conjunctions! It is now a pair $(head, body)$. This is extremely difficult to follow and relate to the previous material. A conjunction is not obtained from the frontier? Apparently, not... Examples also do not show complex terms. Is there any normal form for equations?

Page 13, Line 1: What is the form of disequations?

Page 13, normalization of the conjunction: Now appears the definition of $GoalVars$. But, what is $ImpVars$? The way it is stated apparently it is the same as $RelVars$. But afterwards $ImpVars$ is defined as the set of variables in $GoalVars$ and variables that appear in $\bar{I}$.

Page 13, end of section 2.2 format better the example.

Page 14: Write implementation details in capital letter.

Page 14, last paragraph before 2nd step: "frequent" instead of "frecuent"

Page 16, 3rd paragraph: "infinitely many answers" instead of "infinitely many answer".

Page 17, line 4: "join" instead of "joint".

Page 18, line 1: "`\==`" instead of "`/==`"

Page 18, last line of table the last constraint is wrong. There is an s missing.

Page 20, constraint simplification: what are frontier variables?

Page 20, last line: Illustrate the method with examples. Furthermore, it might not be clear to the reader that $\bar{x}$ is obtained by renaming of variables.

Page 24, 2nd paragraph of section 5.1: I do not understand the meaning of "the negation of failure that is success"?