**Coinductive Logic Programming and its Application to Boolean SAT**

Co-induction has been recently introduced into logic programming and shown to have interesting applications to modeling and reasoning about infinite processes and objects. Co-inductive logic programming has also been extended with negation resulting in yet more applications. The most interesting of these applications is leveraging co-induction and negation to obtain goal-directed strategies for executing answer set programs. Answer Set Programming is a powerful paradigm for performing non-monotonic reasoning within logic programming. In the given paper they show yet another application of co-inductive logic programming, namely, to elegantly obtaining Boolean SAT solvers. We show how co-LP extended with negation can be used to obtain Boolean SAT solvers. We contrast this method of obtaining Boolean SAT solvers to the one based on using answer set programming.

SLD resolution (for definite LP) extended with the co-inductive hypothesis rule is called co-SLD resolution (Simon et al. 2006). Co-SLDNF resolution, devised by the authors, extends co-SLD resolution with negation. Essentially, it augments co-SLD with the negative co-inductive hypothesis rule, which states that if a negated call not(p) is encountered during resolution, and another call to not(p) has been seen before in the same computation, then not(p) co-inductively succeeds. First we define a (well-formed) query (or current resolvent ) , and the notion of positive or negative context of a literal occurring in it. Note that nt(A) below denotes co-inductive “not” of A.

Answer Set Programming (ASP) and its stable model semantics have been successfully applied to elegantly solving many problems in non-monotonic reasoning and planning. Answer Set Programming is a declarative logic programming language. Its basic syntax is of the form: Lo :- L1, … , Lm, not Lm+1, …, not Ln. (1) where Li is a literal where n >= 0 and n >=m.

In the paper given they showed how co-SLDNF resolution can be used to elegantly develop Boolean SAT solvers. The SAT solvers thus obtained are simpler and more elegant than solvers realized via ASP. We also discussed how co-SLDNF resolution can be employed to obtain goal-directed execution mechanisms for answer set programs. Our future work is directed towards making the implementation of both co-ASP and co-SAT more efficient so as to be competitive with the state-of-the-art solvers for ASP and SAT.