Constructing provably correct logic programs

Włodzimierz Drabent

September 15, 2023

Section 2.4 (p. 9 and p. 10) from
W. Drabent. "Logic + control: On program construction and verification".
Theory and Practice of Logic Programming 18(1):1-29, 2018.
DOI: 10.1017/S1471068417000047,

This note presents a systematic approach to construct logic programs that are provably correct and semi-complete. (Semi-complete means, roughly, complete whenever the program terminates.)

Basic explanations: A *specification* is a set of ground facts. An atom H is *covered* by a rule (i.e. a program clause) C w.r.t. a specification S if H is the head of a ground instance $H \leftarrow B_1, \ldots, B_n$ of C, such that the atoms $B_1, \ldots, B_n$ are in S. The sufficient condition for program correctness w.r.t. S is that for each ground instance $H \leftarrow B_1, \ldots, B_n$ of each rule C of the program, if the atoms $B_1, \ldots, B_n$ are in S then H is in S.

It often happens that the specifications for completeness and correctness differ. For instance $append([\,], a, a)$ may not be in the specification for completeness (as it is not required to be computed), but in the specification for correctness (as it is acceptable as an answer).

2.4 Program construction

The presented sufficient conditions suggest a systematic (informal) method of constructing programs which are provably correct and semi-complete. A guiding principle is that the program should satisfy the sufficient condition for semi-completeness. The construction results in a program together with proofs of its correctness and semi-completeness.

Assume that specifications S_{compl} and S_{corr} are given for, respectively, completeness and correctness of a program P to be built. For each predicate p occurring in S_{compl}, consider the set $S_p = \{\, p(\bar{t}) \mid p(\bar{t}) \in S_{\text{compl}} \,\}$ of the specified p-atoms from the specification. To construct a procedure p of P, provide rules such that

(1) each atom $A \in S_p$ is covered w.r.t. S_{compl} by some rule, and
(2) each rule satisfies the sufficient condition for correctness w.r.t. S_{corr} of Theorem 5.

(In other words, the first requirement states that the constructed procedure satisfies the sufficient condition for semi-completeness.) The constructed program P is the union of the procedures for all p from S_{compl}. It satisfies the conditions of Theorems 5, 9. Thus, P is correct w.r.t. S_{corr} and semi-complete w.r.t. S_{compl}.

In practice, semi-completeness is not sufficient. The actual task is to obtain a program which is complete; also in most cases, the program should terminate for the intended class of initial queries. So the constructed program rules should additionally satisfy some sufficient condition for completeness (like that of Theorem 13) or for

termination (like the program being recurrent). In this way, the method of the previous paragraph may be augmented to ensure not only correctness and semi-completeness, but also completeness.

The approach presented here will be used throughout the program constructions presented in this paper.