ON COMPLETENESS OF THE INDUCTIVE ASSERTION METHOD FOR LOGIC PROGRAMS

May 1988

Włodzimierz Drabent

Institute of Computer Science
Polish Academy of Sciences
Box 22
00-901 Warszawa PKiN
telex: 813556 coan pl

ABSTRACT: Completeness of the inductive assertion method for logic programs is shown under an assumption that the assertion metalanguage of the method is expressive enough.

The inductive assertion method (IAM) for logic programs was introduced in [DM1]. [DM2], [DM3] are improved and updated versions of the initial report. These papers do not discuss the completeness of the method. In this paper a kind of relative completeness is shown.
IAM is complete provided that the metalanguage of assertions is expressive enough. A sufficient condition for this is that any recursively enumerable set (relation) can be described by a precondition (postcondition) in this language. The expressive power of the language depends on the set of predicates and functions used.

Below we use the definitions of [DM3] (or [DM1], [DM2]). The presentation is informal, as the metalanguage of assertions of IAM is not defined formally.

IAM is a method of proving (partial) correctness of asserted logic programs. If the verification condition (VC) of the method (Theorem 4.3 of [DM3]) is satisfied then the program is correct. VC is expressed in semantic terms. It refers to the facts true in a fixed interpretation domain. The domain is that of terms with relations of equality, subterm etc and functions of term construction, term selection etc. As natural number arithmetics can be modelled in this domain, there does not exist a finite set of axioms and rules of inference that make possible deriving all (and only) these facts.

The question of completeness concerns the scope of applicability of the method (under which circumstances it can be succesfully applied) [Apt p. 437, 2.7 par.1]. The notion of completeness of a proof method refers to the ability of the method to prove any fact which is expressible in the language of the method provided the fact is true in the class of interpretations under consideration. Of course it is not the case that any correct asserted program satisfies VC. Thus not every correct asserted program can be proven correct using the method.

Example of such program:

```
<- p(X), q(X).
p(7).
```

p : **pre true; post true**
q : **pre** $'q_1=7$; **post true**

Obviously, the verification condition does not hold for the goal clause due to a too weak postcondition for p. Changing the postcondition for p to $p_1'=7$ we obtain an asserted program that can be proven correct using IAM.

Let us introduce some definitions. An assertion p : **pre** E; **post** F is ***stronger*** than p : **pre** E_1; **post** F_1 iff E implies E_1 and F implies F_1 (in the fixed interpretation of functors and predicate symbols of the metalanguage of assertions as described in [DM3], section 2).

A set **A** of assertions is stronger than a set **B** of assertions if (1) the assertions in **A** (B) are for distinct predicate symbols and (2) for every assertion p : **pre** E_1; **post** F_1 of **B** there exists an assertion p : **pre** E; **post** F of **A** such that p : **pre** E; **post** F is stronger than p : **pre** E_1; **post** F_1.

Now, IAM is complete in the following sense provided that the metalanguage of assertions is expressive enough:

PROPOSITION
Let $P \cup B$ be a correct asserted program where P is a set of clauses and B a set of assertions. Then there exists a set of assertions A stronger than B such that $P \cup A$ satisfies the verification condition of IAM. [Thus the correctness of $P \cup A$ is provable by IAM].

Proof

Let $\mathcal{D}er$ be the set of all the SLD-derivations of P. For each p take $A(p)$ - the set of all call patterns (ie. selected subgoals) of p occuring in $\mathcal{D}er$ and $R(p)$ - the set of all call-success pairs of $\mathcal{D}er$. Now let $E(p)$ be a precondition satisfied exactly by elements of $A(p)$ and let $F(p)$ be a postcondition satisfied exactly by elements of $R(p)$. If $E(p)$ and $F(p)$ exist in the metalanguge of assertions then

A = { p : **pre** $E(p)$; **post** $F(p)$ | p occurs in P }

is the required set of assertions. It remains to show that $P \cup A$ satisfies verification condition VC.

Note first that if $(a, a\sigma)$ satisfies its postcondition (from A) then there exists an SLD-derivation $\mathcal{D}$ of P such that $\leftarrow a, G$ and $\leftarrow G\sigma$ are goals of $\mathcal{D}$ (where **G** is an atom sequence). Hence for any atom sequence H, for P with the initial goal exchanged for $\leftarrow a, H$ there exists an SLD-derivation $\mathcal{D}'$ such that $\leftarrow H\sigma$ is a goal of $\mathcal{D}'$.

VC consists of three implications. Their premises refer to an atom b and a set of atom pairs that satisfy their postconditions. From the reasoning above and the definitions of A, R and of a valuation sequence it follows that there exists an SLD-derivation for P where b is a call pattern and these pairs are call-success pairs. (We skip details of the comstruction.) The conclusions of these implications refer to an atom or an atom pair. For each implication this atom (pair) is a call pattern (call-success pair) of the SLD-derivation exemplified above. Thus the atom (pair) satisfies its pre- (post-) condition and VC holds.

End of proof.

As $A(p)$ and $R(p)$ are recursively enumerable, for $E(p)$ and $F(p)$ to exist it is sufficient that any recursively enumerable set is expressible in the metalanguge.
Note that the reasoning above remains valid also for programs with countably many goal clauses. (IAM is often used for a program with a class of goals).

FUTURE WORK

The sufficient condition that any recursively enumerable set and relation can be described in the metalanguage is rather strong. An interesting question is weakening this condition by investigating for which assertion (meta-) languages IAM is complete. This would lead to a kind of completeness in the sense of Cook [Apt]. A related problem is: for a given program P and a true assertion for a predicate symbol in P find assertions for the remaining predicate symbols such that VC holds (thus the correctness of the resulting asserted program is provable by IAM).

REFERENCES

[Apt] Apt, K.R. Ten years of Hoare's logic: a survey - part 1. *ACM ToPLaS*, 3, 4 (October 1981), 431-483.

[DM1] Drabent, W., Małuszyński, J. Proving run-time properties of logic programs, Linkoping University, Research report LITH-IDA-R-86-23, July 1986.

[DM2] Drabent, W., Małuszyński, J. Inductive assertion method for logic programs, In *Lecture Notes in Computer Science*, vol. 250: *Proceedings TAPSOFT '87*, vol. 2, 167-181.

[DM3] Drabent, W., Małuszyński, J. Inductive assertion method for logic programs, *Theoretical Computer Science*, vol. 59 (1988) 133-155.