COMMENTS ON "GENERAL FAILURE OF LOGIC PROGRAMS"

JOXAN JAFFAR,* JEAN-LOUIS LASSEZ, AND MICHAEL J. MAHER**

The paper [1] purports to present a classification of the general failure sets of logic programs and a simple proof of the theorem on the soundness and completeness of the negation-as-failure rule. In this note we clarify some conflicting terminology between [1] and the papers [2, 3] to which it predominantly refers. Our main purpose, however, is to point out major errors, in particular, one in the proof of the above mentioned theorem.

INTRODUCTION

In [1], results are claimed to be established about general failure sets, ground general failure sets and negation as failure. In a first part of this note we show that crucial definitions used in [1] are not equivalent to those in the literature, even though the notations and terminology are the same. This confusion invalidates the claims that a conjecture from [3] has been solved and that new proofs of classical results have been found. In a second part we show that in any case the main results of [1] are wrong. Similar confusions and erroneous results are to be found in a follow up paper by the same author [7].

NOTATION AND TERMINOLOGY

The major conflict between [1] and [2, 3] lies in the definition of general failure sets. In what follows, we use underscores to distinguish between notation from [1] which is in conflict with [2, 3] and the symbols $P$ and $E$ to denote an arbitrary logic
program and equality theory respectively. In [2, 3], two kinds of general failure sets were introduced:

(a) \( GF(P, E) \), the general failure set, is the set of those ground atoms all of whose fair \((P, E)\)-derivations are finitely failed. No inductive definition for this set was presented.

(b) \( GGF(P, E) \), the ground general failure set, is the set of ground atoms all of whose ground fair \((P, E)\)-derivations are finitely failed. An inductive definition of this set was given as the union, over all ordinals \( \alpha \), of a transfinite sequence of sets \( GGF_0(P, E), GGF_1(P, E), GGF_2(P, E), \ldots \).

It was proven that these two sets are, in general different.

In [1], a predicate \( F(B, n_i, A_i, k) \), where \( B \) is a ground atom and \( k \leq \omega \), was defined. In case \( k = \omega \), the defining sentence is infinite. The sets \( GF(P, E) \) and \( GGF(P, E) \) in [1] can then expressed as follows:

(a) \( GF(P, E) = \{ B : F(B, n_i, A_i, 1) \} \). An inductive definition of the sets \( GF_k(P, E) \), \( k < \omega \), was given.

(b) \( GGF(P, E) = \{ B : F(B, n_i, A_i, k) \) for some \( k \leq \omega \). \( \) An inductive definition of the sets \( GGF_\alpha(P, E) \) was given exactly as in [2, 3].

We now observe that

\[ GF(P, E) \neq GGF(P, E) \quad \text{and} \quad GF(P, E) \neq GGF(P, E). \]  

A simple example will prove both the above inequations.

**Example 1.** Take the program \( P \):

\[
p(a) \leftarrow q(X), \\
q(f(X)) \leftarrow q(X),
\]

consider \( E = \{ \} \), i.e. as in pure PROLOG, and let \( HB \) denote the Herbrand base for this program. In this case, \( GF(P, E) = HB - \{ p(a) \} \), \( GF(P, E) = HB \), \( GGF(P, E) = HB \), and \( GGF_\alpha \), the union of \( GGF_\alpha(P, E) \), over all \( \alpha \), is \( HB \).

It follows immediately from the definitions in [1] that \( GF_k(P, E) = GGF_{\omega + k}(P, E) \), and so

\[ GF(P, E) = GGF_{2\omega}(P, E) \]  

A further difference in terminology in [1] from [2, 3] lies in the definition of \((P, E)\)-derivation sequences. In [2, 3], a derivation sequence was defined as a sequence of goals, each of which may contain variables. Goals are derived from preceding ones using \( E \)-unifiers. Ground derivation sequences were then defined to be derivation sequences where only ground \( E \)-unifiers are employed. In [1], derivation sequences are based upon sequences of literal lists of ground instances of clauses. A "single derivation sequence" corresponds to a ground derivation sequence in [2, 3]. A "\((P, E)\)-derivation sequence", for a single goal, encapsulates all single derivation sequences for that goal. The important point to note is that the operational model in [1] differs from that in [2, 3]; in particular:

derivation sequences in [1] correspond only to the ground derivation sequences in [2, 3].
We summarize this section as follows. From (1) on, none of the general failure sets \( GF(P, E) = \bigcup GF_o(P, E), GFF(P, E), \) and \( \bigcup GGF_a(P, E) \) defined in [1] correspond to \( GF(P, E) \) in [2,3]. Thus, for example, the problem of an inductive definition for \( GF(P, E) \) has not been addressed at all. \( GF(P, E) \), by definition, is concerned only with ground derivations. Equation (2) shows that \( GF(P, E) \) is, in fact, determined solely by the \( GGF_a(P, E) \). In contrast, no such relationship (2) exists between \( GF(P, E) \) and the \( GGF_a(P, E) \), that is, there is no ordinal \( \alpha \) such that \( GF(P, E) = GGF_a(P, E) \) for all \( P \) and \( E \). On the issue of negation as failure, Points (1) and (3) show that, in [1], the notions of failure and negation as failure are different from [2,3]. For example, when \( E = \{ \} \) and we are considering programs in pure PROLOG, the notion of failure in [1] is not equivalent to the notion of failure as used by Clark [4].

TECHNICAL MATTERS

We address three major technical points in [1], the first of which concerns Proposition 3. This proposition is wrong; as stated, the proposition implies that \( GGF_0(P, E) = GGF_1(P, E) \), and this in turn implies that \( GGF_0(P, E) = GGF_a(P, E) \) for any ordinal \( \alpha \). Two possible alternatives to this proposition are:

**Proposition 3a.** For any limit ordinal \( \alpha < \omega^2 \) and ground atom \( B \), we have \( B \in GGF_a(P, E) \) iff \( F(B, n_i, A_i, k, \alpha) \), where \( k_\alpha \) is the number of limit ordinals less than \( \alpha \).

**Proposition 3b.** For any ground atom \( B \),

(i) for integers \( k \geq 0 \) and \( h > 0 \), \( B \in GGF_{k\omega + h}(P, E) \) iff \( S^h(B) = \{ \} \) or \( \forall C \in S^h(B), F(C, n_i, A_i, k) \), and

(ii) for \( k > 0 \), \( B \in GGF_{k\omega}(P, E) \) iff \( F(B, n_i, A_i, k - 1) \).

We further note that Proposition 3b(ii) is equivalent to Proposition 3a.

The second point we raise concerns the meaning of the statement \( F(B, n_i, A_i, k) \) (cf. p. 162 in [1]) when \( k = \omega \). No definition for the meaning of this infinite sentence is given in [1]. It follows from Proposition 3a that

\[
\begin{align*}
F(B, n_i, A_i, 0) & \iff B \in GGF_0(P, E), \\
F(B, n_i, A_i, 1) & \iff B \in GGF_{2\omega}(P, E), \\
F(B, n_i, A_i, 2) & \iff B \in GGF_{3\omega}(P, E), \\
\vdots \\
F(B, n_i, A_i, i) & \iff B \in GGF_{(i+1)\omega}(P, E), \\
\vdots 
\end{align*}
\]

One might be led to believe, from this sequence, that

\[
F(B, n_i, A_i, \omega) \iff B \in GGF_{\omega^2}(P, E)
\]

However, it is easily ascertainable that Proposition 4 in [1] (which is given without a
proof) is correct if and only if

\[ F(B, n, A, \omega) \iff B \in \text{GGF}_{\Omega}(P, E), \]

where \( \Omega \) is the least nonconstructive ordinal. For this reason, and the fact that Theorem 1, one of the two main results in [1], is equivalent to Proposition 4, a formal definition for the meaning of \( F(B, n, A, \omega) \) is required.

Our third and final point concerns the other main result in [1], Corollary 2. Assuming Theorem 1 in [1], this corollary is wrong. The following is a simple counterexample: take \( P = \{ r(c) \leftarrow, s(d) \leftarrow \} \) and \( E = \{ \} \), so that \( P \) is a pure PROLOG program. The ground atom \( r(d) \in \text{GGF}(P, E) \), but \( \neg r(d) \) is not a logical consequence of \( (P^*, E) \). The corollary does not hold for this example in particular because \( E \) does not forbid \( c = d \). In general, there is, in [1], no notion of completeness of the underlying equality theory \( E \) as in, for example, [4] (which uses a particular set of axioms) and [2, 3] (which use unification-complete equality theories).

Corollary 2 in [1] remains wrong even when the equality theory \( E \) is complete in the sense of [2, 3]. For example, let \( E \) be the equality theory given by [4] so that \( E \) is unification-complete [2, 3]. A counterexample can then be obtained by using yet again the pure PROLOG program in Example 1: \( p(a) \in \text{GGF}(P, E) \) but \( \neg p(a) \) is not a logical consequence of \( (P^*, E) \) (cf. problem section, Chapter 3 in [5]). The corollary does not hold for this example in particular because only \( E \)-models are considered. If logical consequence is considered only with respect to this restricted class of interpretations, i.e. the corollary reads

Corollary 2a. \( (P^*, E) \models E \neg B \iff \overline{B} \in \text{GGF}(P, E) \), then this corollary, which follows trivially from results in [3], does not address the issues of soundness and completeness of the negation-as-failure rule in the sense of [2, 3, 4, 5, 6].

REFERENCES