A Fix-Point Semantics for Rule-Base Anomalies

Du Zhang, California State University, USA

ABSTRACT

A crucial component of an intelligent system is its knowledge base (KB) that contains knowledge about a problem domain. KB development involves domain analysis, context space definition, ontological specification, and knowledge acquisition, codification, and verification. KB anomalies can affect the correctness and performance of an intelligent system. In this article, we describe a fix-point semantics for a KB that is based on a multi-valued logic. We then use the fix-point semantics to provide formal definitions for four types of KB anomalies: (1) inconsistency, (2) redundancy, (3) incompleteness, and (4) circularity. We believe such formal definitions of KB anomalies will help pave the way for a more effective KB verification process.

Keywords: fix-point semantics; knowledge base anomalies; multi-valued logic

INTRODUCTION

Computing plays a pivotal role in our understanding of human cognition (Pylyshyn, 1989). The classical cognitive architecture for intelligent behavior assumes that both computers and minds have at least the following three distinct levels of organization (Pylyshyn, 1989): (1) The semantic level or the knowledge level where the behavior of human beings or appropriately programmed computers can be explained through the things they know and the goals they have. It attempts to establish, in some meaningful or even rational ways, connections between the actions (by human or computer) and what they know about their world. (2) The symbol level where the semantic content of knowledge and goals is assumed to be encoded through structured symbolic expressions. It deals with representation, structure, and manipulation of symbolic expressions. (3) The physical or biological level where the physical embodiment of an entire system (human or computer) is considered. It encompasses the structure and the principles by which a physical object functions.

Pylyshyn’s (1989) cognitive penetrability criterion states that “the pattern of behavior can be altered in a rational way by changing
subjects’ beliefs about the task” (Pylyshyn, 1989). It is the subjects’ tacit knowledge about
the world, not the properties of the architecture
that enables such behavior adjustment.

The hallmark of a knowledge-based sys
stem is that by design it possesses the ability
to be told facts about its world and to alter its
behavior accordingly (Brachman & Levesque,
2004). It exhibits the property of cognitive
penetrability.

Today, knowledge-based systems not only
play an important role in furthering the study
in cognitive informatics (Chan, Kinsner, Wang,
& Miller, 2004; Kinsner, Zhang, Wang, & Tsai,
2005; Patel, Patel, & Wang, 2003; Wang, 2002,
2007; Wang, Johnston, & Smith, 2002; Wang &
Kinsner, 2006), but also have found their way
into so many problem domains (Cycorp, 2006)
and have been utilized to generate numerous suc
cessful applications (IBM, 2006; Ross, 2003).
A crucial component of an intelligent system or a
knowledge-based system is its KB that contains
knowledge about a problem domain (Brachman
& Levesque, 2004; Fagin, Halpern, Moses, &
Vardi, 1995; Levesque & Lakemeyer, 2000).
KB development involves domain analysis,
context space definition, ontological specifica
tion, and knowledge acquisition, codification,
and verification (Zhang, 2005).

When developing a KB for an application,
it is important to recognize the context under
which we formulate and reason about domain-
specific knowledge. A context is a region in
some n-dimensional space (Lenat, 1998). In
a KB development process, domain analysis
should result in identification of the region
of interest in the context space. Specifying a
context entails specifying or locating a point
or region along each of those n dimensions.
Once the context (or contexts) for a problem
domain is identified, ontological development
is in order. An ontology is a formal, explicit
specification of a shared conceptualization
(Chandrasekaran, Josephson, & Benjamins,
1999; Gomez-Perez, Fernandez-Lopez, &
Corcho, 2004; O’Leary, 1998). After the concep
tualization is in place, knowledge acquisition,
codification, and verification can be carried out
to build the KB for some application. Inevitably,
there will be anomalies in a KB as a result of
existing practices in its development process.
KB anomalies can affect the correctness and
performance of an intelligent system, though
some systems are robust enough to perform
rationally in the presence of the anomalies. It
is necessary to define KB anomalies formally
before identifying where they are in a KB and
deciding what to do with them.

In this article, our focus is on formal defini
tions of KB anomalies and on the issue of how
to identify them. Our attention is on rule-based
KB. A rule-based KB has a set of facts that is
stored in a working memory (WM) and a set
of rules stored in a rule base (RB). Rules repre
sent general knowledge about an application
domain. They are entered into an RB during
initial knowledge acquisition or subsequent
KB updates. Facts in a WM provide specific
information about the problems at hand and
may be elicited either dynamically from the
user during each problem-solving session, or
statically from the domain expert during knowl
dge acquisition process, or derived through
rule deduction.

We assume that rules in a KB have the
following format: $P_1 \land ... \land P_n \rightarrow R$, where $P_i$ are the conditions (collectively, the left-hand side [LHS] of a rule), R is the conclusion (or right-hand side [RHS] of a rule), and the symbol “$\rightarrow$” is understood as the logical implication. The $P_i$s and R are literals. If the conditions of a rule instance are satisfied by facts in WM, then the rule is enabled and its firing deposits its conclusion into WM.

A fact is represented as a ground atom. It
specifies an instance of a relationship among
particular objects in the problem domain. WM
contains a collection of positive ground atoms
that are deposited through either assertion (ini
tial or dynamically), or rule deduction.

A negated condition $\neg p(x)$ in the LHS of a
rule is satisfied if $p(x)$ is not in WM for any $x$.
A negated ground atom $\neg p(a)$ in the LHS of a
rule is satisfied if $p(a)$ is not in WM. A negated
conclusion $\neg R$ in the RHS of a rule results in
the removal of R from WM, when the LHS of