CANDID Specification of Commercial and Financial Contracts: A Formal Semantics Approach to Knowledge Representation, Part III: CANDID Specification of Financial Concepts

Ronald M. Lee

Florida International University (FIU), Miami, FL, USA, r.lee@fiu.edu

Abstract. The formal language CANDID is presented as a knowledge representation formalism for artificially intelligent decision support systems. The language is specifically oriented to representation of concepts in finance, commerce and administration. Later parts of the paper demonstrate the application of CANDID to explication of corporate entities and contractual objects, as well as to various concepts in elementary finance.

1 Introduction

In Part I, the formal descriptive language CANDID was developed. In Part II, this was applied to the description of the principal entities of economic activity, what we called economic actors, and economic objects. In this part, we extend the application of CANDID to consider the processes of economic activity itself, in describing the concepts of elementary finance, i.e. common types of transactions and financial instruments. We find this domain to be not only a fairly central and important one to understanding commercial activity more broadly, but also reasonably representative of the classes of conceptual problems likely to arise in efforts to formalize other aspects of business. We thus believe that analogous analyses could be applied for instance to financial accounting, cost accounting, tax law, contract law, regulatory law, etc. Again, we want to emphasize that CANDID is proposed as a *framework* for formalizing business theory, but is not intended as a theory itself. The discussion here is thus meant to be only illustrative, attempting to capture what we see as the ordinary usage and understanding of basic financial terminology and concepts. Various contemporary theories of accounting, finance and economics might therefore disagree with aspects of the analysis here. (The only responsibility we would claim for CANDID is to explicate this disagreement.)

As a general guide to what concepts should be included here, we made use of *Mathematics of Finance* [Aye63], a beginning level college primer. This is likewise suggested as an elementary background reference.

2 Additional Definitions, Notational Conventions

In Part II the concepts of an *economic actor* and *economic object* were developed. Informally, an economic actor is a legally able person or organization (proprietorship, partnership or corporation) while an economic object is a physical object (excluding persons), a contractual object (e.g. stock, bonds, licenses), a monetary object (cash or demand deposit checks) or an information object (e.g. textual materials, computer data and programs). In addition, two two-place relations between economic actors and economic objects were assumed. *OWN* (for ownership) and *POSS* (for possession). These have the following associated axioms:

OWN(x,z) → ECON-ACTOR(x) & ECON-OBJ(z).

POSS(x,z) → ECON-ACTOR(x) & ECON-OBJ(z).

Also, the notation $$ is used to indicate U.S. currency in cash or check form. E.g.

$$(m) = 158.32

indicates that the object m is a sum of money totaling $158.32. As in the earlier parts, parentheses are used for functional application arguments for predicates and functions), while square brackets are used for syntactic disambiguation. Also as previously, predicates may indicate states, changes or actions. As a visual aid, we append "!" to predicate names for changes and "!!" to names of actions. Thus, as in Part II, we have the following definitions of changes and action relating to ownership and possession.

OCHANGE!(x,y,z) ::= OWN(x,z) T OWN(y, z)
PCHANGE!(x,y,z) ::= POSS(x,z) T POSS(y,z)

OCHANGE! indicates a change in ownership of z from x to y.
PCHANGE! indicates an analogous change of possession.

OGIVE!!(x,y,z) ::= OWN(x,z) T [OWN(y, z) I(x) OWN(x,z)]
OTAKE!!(x,y,z)) ::= OWN(x,z) T [OWN(y, z) I(y) OWN(x,z)]
PGIVE!!(x,y,z)) ::= POSS(x,z) T [POSS(y,z) I(x) POSS(x,z)]
PTAKE!!(x,y,z)) ::= POSS(x,z) T [POSS(y,z) I(y) POSS(x,z)]

OGIVE!! indicates a change of ownership from x to y initiated by x, whereas *OTAKE*!! indicates the same change of ownership, but initiated by y. *PGIVE!!* and *PTAKE!!* are similarly defined for possession. One additional definition