Introduction

David J. Pym

Informatics may be defined as *the science of the structure, complexity and communication of information.* As such, informatics is concerned with the study of the structure, behaviour, interactions and construction of natural, artificial and abstract systems. It has philosophical, mathematical, computational and social aspects. It has emerged in the wake of the (electronic) computer and the central function of the computer, the transformation of information, is its unifying notion.

Logic is usually defined as the *science of reasoning.* We suggest that a better definition is that it is the study of the *structure of information.* It has its roots in *grammar* and the *semantics of natural languages* but has been of central importance in *mathematics* and its *foundations.* With the growth of the computing-driven sciences, *i.e.*, of informatics, logic has developed in new and challenging ways. Building on the established model theory and recursion theory, informatics has driven new emphases on proof theory and constructivity.

Entirely new concerns have arisen, however, with the following being leading examples:

- Program logic: We want to reason about the behaviour of programs. The leading example of program logic is Hoare's logic [Apt, 1989]. The basic propositional assertions are

  PRE-CONDITION **Program** POST-CONDITION.

  The PRE-CONDITION and POST-CONDITION are assertions in a predicate logic and **Program** is a procedure written in a (typically imperative) programming language. The pre- and post-conditions may, for example, be assertions about the computer memory used by the procedure.

- Processes and Nets: Computational systems consist in networks of communicating devices. Each device, such as a processor, a computer, a printer, a scanner or a user, must interact with its peers. For this interaction to occur, the devices must both exchange and process information. To facilitate reasoning about the behaviour of such networks, two leading logical formalisms have been proposed, namely Petri Nets [Reisig, 1998] and Process Calculi [Milner, 1975, Hoare, 1985, Milner, 1989, Milner, 1999]. Nets provide a graphical description of networks and their connectivity of devices whereas process calculi use algebraic methods to model the transmission of information.

- Resources: Whenever a procedure executes, resources are consumed. Resources may, for example, be spatial, such as a computer's memory, temporal (such as CPU cycles) or monetary (such as the coins required to obtain goods from a vending machine). More delicately, resources may also be dynamic, such as processes [Hansen, 1973]. The challenge, the addressing of which is begun herein by the development of **BI**, *the logic of bunched implications*, is to provide a mathematical model which adequately describes such apparently diverse phenomena and to obtain from it a logic which may be used to reason about resources.

- Logical frameworks: When a programmer writes a program he describes to a computer a model of an external phenomenon, *i.e.*, the application. Logically, we may think of the corresponding situation in which one logic (the object-logic) is represented by another (the meta-logic):

Logic	Programming
Object-logic	Application model
Meta-logic	Programming Language

This idea extends the range of mathematical logic from the study of given systems to the study of the representation of families of systems within a given system. Work on this topic began in earnest with the LF logical framework [Harper et al., 1987, Harper et al., 1993, Pym, 1990, Avron et al., 1992, Pym and Wallen, 1991, Pym and Wallen, 1992, Pym, 1995b, Pym, 1996, Pym, 1995a] and continues to be a major topic. However, LF's basis in intuitionistic logic (via the $\lambda\Pi$-calculus) leads to difficulties in representing program logics, such as Hoare's logic, of the kind described above [Mason, 1986]. The reason for this is the failure of the semantics of intuitionistic logic to account for the spatial properties of the resources to which

Hoare's logic inherently refers. A logical framework, RLF, based not on intuitionistic logic but rather on a substructural logic which has a semantics based on resources, provides a better analysis of program logics.

We begin with this introductory chapter in which we survey the background to our work on **BI**. Starting from a semantic view of logical consequence in terms of truth we go on to consider a range of formal calculi for constructing proofs. We then consider how the semantic view of consequence is affected if "possible worlds" are interpreted as (and constructed as) "resources". We move on to consider the relationship between the construction of proofs and the use of resources. We conclude with a sketch of the idea of logical frameworks, the study of the representation of systems of logic in a formal meta-logic, and consider the possibilities for a semantics based on resources.

Consequences, Truth and Proof

Logic may be seen as the study of *consequences*, *i.e.*, assertions that the truth of a given proposition follows from the truth of a given *collection* of *propositions*. Propositions are *declarative statements*. We can give a simple definition, as described in Hodges [Hodges, 1993], as follows:

> A proposition is that situation which is described by an English phrase which may be substituted for X in
>
> *It is the case that* X.
>
> so as to give a grammatically correct English sentence.

Examples are phrases like "the earth is flat", "the sun orbits the earth" or "I have enough coins to buy a chocolate bar from the vending machine." Barwise [Barwise and Perry, 1983, Barwise, 1989], and others, have developed *situation theory* in order to further analyse this linguistically-derived perspective on the notion of proposition in terms of *situations* and *infons*. Devlin [Devlin, 1990] provides a thorough description of these ideas. Mathematically, propositions are denoted by the *formulæ* of a formal language.

Logic is about more than propositions, however; it is also about reasoning. In classical logic (**CL**), intuitionistic logic (**IL**) and linear logic (**LL**), the basic notion of reasoning is captured by the idea of a *consequence relation* [Tarski, 1956, Scott, 1974, Avron, 1991]

$$\phi_1; \dots ; \phi_m \vdash \psi_1; \dots ; \psi_n$$

between finite sequences of propositions. It should be read as follows:

If we have all of the ϕs, then we have at least one of the ψs.

A common restriction is to the case in which $n = 1$. Formally, a consequence relation on set of formulæ is a binary relation $\vdash$ between finite sequences of formulæ such that:

1 Reflexivity: for every formula ϕ, $\phi \vdash \phi$;

2 Transitivity (or Cut): if $\Gamma \vdash \Delta; \phi$ and $\phi; \Gamma' \vdash \Delta'$, then $\Gamma; \Gamma' \vdash \Delta; \Delta'$.

Additional axioms which may be taken include:

3 Exchange: if $\Gamma \vdash \Delta$, then $\rho(\Gamma) \vdash \sigma(\Delta)$, for permutations ρ and σ;

4 Weakening: if $\Gamma \vdash \Delta$, then $\Gamma; \Gamma' \vdash \Delta; \Delta'$

5 Contraction: if $\Gamma; \Gamma \vdash \Delta$ or $\Gamma \vdash \Delta; \Delta$, then $\Gamma \vdash \Delta$.

Consequence relations are typically realized in two ways, model-theoretically and proof-theoretically. In **CL**, the key semantic notion is *truth*. Explanations of truth in mathematical logic usually begin with the idea of a *truth table* in which a proposition is assigned a *truth value*, 0 (false) or 1 (true). The assignment of truth values is performed by induction on the structure of propositions, connective by connective. For example. the truth table for classical implication is the following:

ϕ	ψ	$\phi \supset \psi$
0	0	1
0	1	1
1	0	0
1	1	1

Here the idea is that we assume, inductively, that we have assignments of truth values for ϕ and ψ — there are four possible combinations — and proceed to assign a value to $\phi \supset \psi$ in each case. We can write similar tables for conjunction, $\wedge$, disjunction, $\vee$, and negation, $\neg$, as follows:

ϕ	ψ	$\phi \wedge \psi$
0	0	0
0	1	0
1	0	0
1	1	1

ϕ	ψ	$\phi \vee \psi$
0	0	0
0	1	1
1	0	1
1	1	1

ϕ	$\neg\phi$
0	1
1	0

Mathematically, we think of such an assignment of truth values, or *model*, as a function

$$\mathcal{I} : \mathbf{Prop} \to \{0, 1\}$$

from the set of propositions to the two-element set. It is then convenient to define $\mathcal{I} \models \phi$, read as "$\mathcal{I}$ satisfies ϕ", by

$$\mathcal{I} \models \phi \quad \text{iff} \quad \mathcal{I}(\phi) = 1.$$

Starting from this point, we can define the notion of *semantic consequence for truth* in a given model $\mathcal{I}$:

$$\begin{array}{lll} \phi_1; \ldots; \phi_m \models_{\mathcal{I}} \phi & \text{iff} & \mathcal{I} \models \phi_i, \text{ for each } 1 \leq i \leq m, \\ & & \text{implies } \mathcal{I} \models \phi. \end{array}$$

A stronger notion is *semantic consequence for validity*, defined as follows:

$$\begin{array}{lll} \phi_1; \ldots; \phi_m \models \phi & \text{iff} & \text{for all } \mathcal{I}, \mathcal{I} \models \phi_i, \text{ for each } 1 \leq i \leq m, \\ & & \text{implies } \mathcal{I} \models \phi. \end{array}$$

These ideas are the very beginning of *classical model theory*, the area of logic which is perhaps mostly deeply integrated with mainstream pure mathematics. By adding *quantifiers*, such as $\forall$, or "for all", and $\exists$, or "there exists", and *theories*, or collections of special symbols and axioms, to the analysis described above, model theory is able to provide a logical study of important mathematical structures. For example, the *model theory of fields* is a major area in its own right. Its axioms include propositions such as

$$\forall x.(x + 0 = x), \quad \forall x.\forall y.\forall z.(x \times (y + z) = x \times y + x \times z)$$

and

$$\forall x.((x \neq 0) \supset \exists y.(x \times y = 1)),$$

where $+$, 0, $\times$ and 1 are *function* symbols used to build the *terms* of the logic, and $=$ is a special *predicate* symbol, taken in addition to the logical connectives and quantifiers. The equality symbol, $=$, is used to build the atomic propositions by *predicating* terms: if s and t are terms we can form the proposition that they are equal by writing $= (s,t)$ or, more simply, $s = t$. Similarly, we write $s \neq t$ as a shorthand for $\neg(s = t)$. From this point of view, a field is a model which satisfies these (and some other) axioms.

Moving on to **IL**, we must adopt a more sophisticated semantics (see [van Dalen, 1983] for an extended discussion). Heyting's formalization of intuitionistic predicate logic, arithmetic and set theory (see also Glivenko and Kolmogorov) [Heyting, 1989, Girard et al., 1989] adumbrated the now-familiar proof-interpretation or BHK semantics but did not provide a framework for a theory of models supporting a notion of truth and a corresponding completeness theorem. The central idea is that the facts

Springer

http://www.springer.com/978-1-4020-0745-3

The Semantics and Proof Theory of the Logic of
Bunched Implications
Pym, D.J.
2002, XLIX, 290 p., Hardcover
ISBN: 978-1-4020-0745-3