An Axiomatisation of Imperatives using Hoare Logic

Miguel Pérez-Ramírez <mperez@essex.ac.uk>
Chris Fox <foxcj@essex.ac.uk>
Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, United Kingdom

December 2002

Abstract

This paper presents an axiomatisation of imperatives using Hoare logic. It accounts for some inferential pragmatic aspects of imperatives. Unlike Jorgensen's, Ross, and Chellas, proposals, rather than assigning truth-values to imperatives, imperatives are evaluated as a relation between the state demanded and the state or circumstances in which the imperative is uttered.

1 Introduction

In general, imperatives are conceived as sentences that are used to issue orders or commands (Radford, 1997; Megginson, 1996; MacFadyen, 1996). This is a broad characterisation, which may include other types of sentences (see Hamblin (1986) for an extensive classification of imperatives).

For simplicity of exposition, we will adopt a syntactic view of imperatives, since this view locates appropriate sentences[1] in a language and allows us intuitively to distinguish them from statements and questions.

Definition 1 *An imperative is a tenseless and subjectless sentence typically used to ask someone to do something or not to do something, and which does not denote a truth-value.*

See Radford (1997: 159-160) for a possible account for the lack of subject in imperatives. Chellas (1971) points out that past tense imperatives are vacuous. The following sentences are considered to be imperatives according to this definition:

We would like to thank to Prof. R. Turner for his help. The first author would like to thank CONACYT-IIE.

[1]Imperatives are a type of sentence. Levinson says "it seems that the three basic sentence types, *interrogative*, *imperative*, and *declarative* are universals, all languages appear to have at least two and mostly three of these" Levinson (1981: p. 242).

Direct imperative:	*Come here!*
Negative imperative:	*Don't do that!*
Conjunction:	*Sit down and listen carefully!*
Disjunction:	*Shut up or get out of here!*
Conditional imperative:	*If it is raining, close the window!*

Requests conveyed by statements and questions such as *I wish you would like you to close the door* (as a request to close the door) and in some contexts *Isn't it hot in here?* (as a request to open the window) are ignored by this definition.

1.1 Paradoxes

Formal semantics has focused largerly on declarative sentences. However some philosophers and logicians have sought to codify the logical properties of imperatives (Jorgensen, 1937; Ross 1941; Segerberg, 1991; von Wright 1968; Chellas, 1971, for example).

Arguably, Jorgensen (1937) initiated the modern discussion about the role of imperatives in inference with his 'dilemma': it seems that imperatives can take part in the so-called practical inferences[2], despite the fact that they cannot be true or false (Ross, 1941; Segerberg, 1990: Weinberger, 1991; Walter, 1996). The following illustrates an example of this dilemma.

If you are cold, put your coat on!
You are cold

Put your coat on!

An inferential pattern is truth preserving, that is, if the premises are true then the conclusion is true. However, in the above example, the first premise and the conclusion cannot be true because they are imperatives. Ross (1941) called this problem "Jorgensen's dilemma."

To address this problem, some proposals, such as that of Jorgensen (1937), attempt to reduce imperatives to statements which are manipulated logically. Conclusions of any practical inference are translated back into imperatives. In this approach *Post the letter!* is translated as *!The letter has to be posted*, using a proposition prefixed by the symbol '*!*' to indicate that the proposition is associated with some imperative.

Huntley (1984) referred to the tendency to use classical logic as the basis for modelling imperatives as the "standard solution." Its key feature is the assumption that the core meaning of imperatives is propositional; something that can be true or false. Typically in such an approach, imperatives are split into two: a descriptive or propositional and a prescriptive part (as in '*!*' above).

[2] A practical inference is an inference where at least one imperative takes part as premise.

Ross (1941) points out that these approaches suffer from paradoxes following from the assumption that classical logic rules and connectives apply between imperatives. Ross illustrates this by showing that from *Post the letter!*—translated as *!The letter has to be posted*—it is possible to infer:

!The letter has to be posted $\vee$ *!The letter has to be burned*

using the rule for introduction of disjunction. Translating back, we obtain the imperative

Post the letter or burn the letter!

This is known as "Ross's counterexample." One of the problems is that a speaker uttered only one order and we are inferring that s/he uttered something else. From the point of view of a hearer, s/he can infer that from a given imperative there is a choice, either to do what has been ordered or to do something else.[3]

Another example is the use of the rule of elimination for conjunction. For instance *Buy oranges and buy apples*! might be translated into

!You have to buy oranges $\wedge$ *!You have to buy apples*

Using the rule of elimination we might infer *!You have to buy oranges*, or *!You have to buy apples* which in turn means that we can infer either *Buy oranges!*, or *Buy apples!* Intuitively this means that from a conjunction of imperatives we are only required to fulfill some of the conjuncts.[4]

1.2 Aim of the paper

In this paper we wish to investigate the logic properties of imperative sentences, and to account for Jorgensen's dilemma, without being affected by Ross's counterexample.

2 Properties of imperatives

We shall assume that imperatives convey requirements.

Definition 2 (Requirements) *According to our definition, an imperative 'asks someone to do something.' This 'asking something' we shall call a* requirement*.*

We shall use the Greek letter ρ (and subscripts) to represent requirements. The examples in Section 1 might be represented as follows:

[3]This problem also arises in Deontic approaches under the name of Free Choice Permission (von Wright, 1968; Kamp, 1973; Meyer, 1996; Meyden, 1996).

[4]The problems described in this section have been reported in detail also in (Pérez-Ramírez, 2000).

Direct imperative: ρ_1 = *Come here!*
Negative imperative: ρ_2 = *Don't do that!*
Conjunction: $(\rho_1; \rho_2)$ = *Sit down and listen carefully!*
Disjunction: $(\rho_1 + \rho_2)$ = *Shut up or get out of here!*
Conditional imperative: $(C \Rightarrow \rho)$ = *If it is raining, close the window!*

An imperative with more than one requirement shall be called *composed*. If the imperative contains only one requirement it shall be called *single*. The Conjunctive and Disjunctive example above are *composed* imperatives and the direct and negative examples are *single* imperatives.

2.1 Imperatives and actions

For Ross (1941), an imperative is satisfied if we have the desired result following an agent's action. For Segerberg, imperatives are prescriptions for actions, and the authority uses them to manipulate the addressee's will. Here it is assumed that both prescriptions of actions (*Come here!*) and prevention of actions (*Don't come here!*) are covered by our definition of requirement. However, we also make a distinction between *requirements* and *implicit actions*. For instance *Buy oranges!* is an imperative expressing one single requirement. Satisfying or complying with this requirement might involve more that one 'implicit action' such as 'taking the bus,' 'going to the market' etc. In some cases, a requirement and its implicit action might coincide.

2.2 Properties

In order to be able to account for paradoxes, we need to consider use of imperatives and when they are considered to be satisfied. This would provide, among other things, a process of legitimisation, which would allow us to recognise under which circumstances an imperative is used appropriately.

2.2.1 Correctness, satisfaction and satisfiability

Following Buvac (1995); De Roeck et al (1991); Manara and De Roeck (1996; 1997) and others, we cast context as a consistent collection of propositions that reflects a relevant subset of agent's beliefs. Interpreting an imperative involves the identification of a state of affairs (context) desired by the speaker. This would allow us to avoid deriving inappropriate conclusions about the expectations of the speaker, such as in Ross' counterexample (Ross, 1941). By knowing what the speaker wants, we might be able to evaluate whether or not the imperative is satisfiable, that is, whether it is possible to achieve what it is demanded.

If we try to entertain imperatives such as *Have three arms!*, we encounter contradictory thoughts. We might think it is a silly sentence, but it does correspond to our syntactic definition of imperatives. We might say that it is a ‘wrong’ use of an imperative. To explain why this use is wrong, we might say that, in normal circumstances, nobody is able to meet this requirement and so it would be silly to express it. It seems that there is at least one pre-condition to use an imperative: it must be possible to satisfy the command. An imperative that meets this conditions shall be referred to as *satisfiable*.

A more feasible imperative “*Close the door!*” looks normal, unless it is issued in a situation where the door is already closed. Along the same lines, compliance with some imperatives is associated with some specific result. For instance, after obeying the imperative *Close the door!* the door will be closed and after *Don't close the door!* an opened door is expected to remain open. From these examples it appears that both pre-conditions and post-conditions are associated with imperatives. Thus, an obvious example of pre-condition for *Close the door!* is $P =$ *The door is open* and the obvious example of post-condition is $Q =$ *The door is closed.* P and Q might be really complex expressions describing states of affairs, including conditions on speakers and hearers abilities.

Imperatives have both objective and subjective aspects. For example, an imperative usually assumes that the speaker possesses some authority over the hearer (See Walter, 1996: p. 170), otherwise it would not be used appropriately. From this point of view, imperatives are *subjective*. The satisfaction of an imperative involves performing a course of actions, these actions have to take place in the objective, physical world. From this point of view imperatives are also *objective*. The previous example would impose the conjunction of at least two pre-conditions as given by the complex expression $P =$ *The door is open* $\wedge$ *Speaker possesses authority over the hearer*, where the former is objective and the latter is subjective.

2.2.2 Correctness

The observations above provide us with some indication of the conditions required when verifying the appropriate use of imperatives. We use the term *correct* to describe a *satisfiable* imperative whose pre-conditions hold when it is issued. Correctness is intended to correspond to the notion of validity used by Ross (1941).

2.2.3 Satisfaction

The notion of correctness includes the requirements that it is possible to reach the state demanded; the imperative must be satisfiable. However it also provides a means to verify satisfied imperatives. That is, it can be use to verify that the post-condition demanded or imposed by the requirement is met. Following Ross (1941), we assume that classical logic is enough to account for the satisfaction of an imperative. For instance, *Close the door!* will be satisfied when the post-condition $Q =$ *The door is closed* is the case.

If $\neg Q$ is the case, then the imperative is not satisfied. *Close the door or the window!* will be satisfied when either *The door is closed* or *The window is closed* is the case or both. Analogously *Close the door and the window!* will be satisfied when both *The door is closed* and *The window is closed* is the case.

2.2.4 Classical connectives and rules are not suitable for imperatives

Ross's counterexample is evidence that the introduction rule for disjunction is not suitable for modelling disjunction of imperatives. Indeed, it is evidence that classical disjunction itself it not appropriate. The same is true of classical conjunction and its rule of elimination. Imperatives can be order dependent. That is, it may not be possible to satisfy one of them before satisfying an earlier one. For example, *Open the envelope and read the letter!* is different from *Read the letter and open the envelope!* This property is subtle. It may be related to implicit event order in declarative discourse. Essentially conjoined dependent requirements or imperatives are not commutative, unlike classical conjunction.

Correctness does not follow classical negation. Consider the example where an employee orders his boss *Do your work!* The imperative is not appropriate because the employee does not have authority to give orders to his boss. On the other hand if the employee utters instead the negation of that imperative *Don't do your work!* It is still not appropriate, again because the employee does not have authority. That is, if an imperative is not appropriate, it does not mean that its negation is. A negated imperative must be considered simply as another imperative which needs to be verified.

Classical logic connectives and rules do not apply between requirements. However, imperatives and statements interact, as is evident in Jorgensen's dilemma, and classical logic can be used to account for the satisfaction of imperatives.

3 Hoare logic

Whilst not immediately concerned with the modelling of imperatives or normative statements, there is some relevant work by Hoare (1969). He sought to develop a framework for reasoning about computer programs, and developed a deductive system for working with imperative languages. This work deserves our attention because it appears to have some characteristics which make it an interesting starting point for a model of imperatives.

Hoare proposes that the initial pre-conditions of successful use can be specified by the same type of general assertion as is used to describe the results obtained on termination. Hoare proposes the notation $P\{\rho\}Q$ to state the required connection between a pre-condition (P), a program (ρ)

and a description of the result of its execution (Q), the post-condition. Hoare interprets this triple as, "*If the assertion P is true before initiation of a program ρ, then the assertion Q will be true on its completion.*" Hoare (1969: p. 577). When there are no pre-conditions, he simply writes $\texttt{true}\{\rho\}Q$. He also uses the notation $\vdash P\{\rho\}Q$ when the relation expressed by the triple can be proved.[5]

Hoare's triple ($P\{\rho\}Q$) encompasses different things. One of the most important is that it provides the formal means to verify the correctness of a program. Furthermore, it establishes a formal relationship between two different kinds of objects, a program (ρ) and assertions (P and Q). It introduces sequencing since P must hold *before* the program ρ, is executed, and Q must hold *after* the execution of ρ.

4 Axiomatisation for imperatives

There are a number of properties shared by commands in imperative programming languages on the one hand and requirements conveyed by imperatives on the other. Based on this similarity, we propose a formalisation of imperatives using Hoare-like rules.

4.1 Representation of imperatives and correctness

Hoare triples are useful for representing imperatives as they provide the means for addressing the fact that imperatives do not have truth-values. Rather than assigning a truth value directly to program ρ, the approach evaluates the relationship given by the triple $P\{\rho\}Q$. That is, by imposing pre/post-conditions, it mixes assertions and programs in inference, and makes a useful distinction between the logical behaviour of programs operations, and reasoning about programs with the classical logic. This is what we need for imperatives, which is why we adopted Hoare triples to represent them. We will assume that at the heart of an imperative lies not a program, but a requirement (a prescription of an action). Thus we might represent the imperative *Close the door!* as follows:

$$\begin{array}{lll} \rho = & \textit{Close the door} & \text{(requirement)} \\ P = & \textit{The door is open} & \text{(pre-condition)} \\ Q = & \textit{The door is closed} & \text{(post-condition)} \end{array}$$

$$P\{\rho\}Q = \textit{ The door is open}\{\textit{Close the door}\}\textit{The door is closed}$$

The interpretation of the expression $P\{\rho\}Q$ for imperatives would read like this "*If an imperative involving the requirement ρ is uttered in an state*

[5]Some authors such as Harel (1979) and Gries (1983) represent Hoare triples using the notation $P \rightarrow [\rho]Q$.

of affairs where P holds, then after the ρ is satisfied, Q will hold in the state of affairs reached."

The term *satisfied* is used here to indicate that the hearer has done what the speaker requested. From now on we adopt the term *correct* to refer to an imperative whose pre-conditions hold in the context in which it is issued, and the post-conditions hold after appropriate action is performed.

We will now explore composed imperatives.

4.1.1 Negation

As a first approximation, the negation of an imperative is simply another imperative. For example *Don't close the door!* may be represented as follows.

$$\begin{aligned} \rho_2 &= \textit{Don't close the door} \\ P_2 &= \textit{The door is open} \\ Q_2 &= \textit{The door is open} \\ P_2\{\rho_2\}Q_2 &= \textit{The door is open}\{\textit{Don't close the door}\}\textit{The door is open} \end{aligned}$$

4.1.2 Conjunction of requirements

We assume that the compositions operators ';' can be used to represent a conjunction of requirements. This accounts for some of the properties we have identified which are problematic in the standard solution.

(I;) If $P\{\rho_1\}Q_1$ and $Q_1\{\rho_2\}Q$ then $P\{\rho_1;\rho_2\}Q$

We may represent imperatives such as *Write a letter and send it!* as follows.

$$\begin{aligned} \rho_1 &= \textit{Write a letter} \quad Q_1 = \textit{There is a letter} \\ \rho_2 &= \textit{Send the letter} \quad Q_2 = \textit{The letter is sent} \\ \neg Q_1\{\rho_1\}Q_1 &= \neg(\textit{There is a letter})\{\textit{Write a letter}\}\textit{There is a letter} \\ \neg Q_2\{\rho_2\}Q_2 &= \neg(\textit{The letter is sent})\{\textit{Send the letter}\}\textit{The letter is sent} \end{aligned}$$

The rule (I;) would allow us to infer the correctness of $\neg Q_1\{\rho_1;\rho_2\}Q_2$, but not the correctness of $\neg Q_2\{\rho_2;\rho_1\}Q_1$, (where the letter is sent before it is written).

The operator ';' models a conjunction of requirements, and the rule (I;) accounts for the property of order dependence.

4.1.3 Conditional requirements

The imperative *If it is raining then close the door!* may be expressed as

$$P\{\phi \Rightarrow \rho\}Q$$

where

$$\rho = \textit{Close the door} \quad \phi = \textit{It is raining} \quad P = \textit{The door is open}$$

Here Q must cover not only the post-condition of ρ when ϕ holds, but also the case when ϕ is false and it is not necessary to satisfy ρ. That is, $Q = \textit{The door is closed} \vee \neg(\textit{The door is closed})$. The expression $P\{\phi \Rightarrow \rho\}Q$ covers the two following cases according to the value of ϕ.

If ϕ is the case: $(P \wedge \phi)\{\rho\}(\textit{The door is closed})$
If $\neg\phi$ is the case: $(P \wedge \neg\phi) \rightarrow \neg(\textit{The door is closed})$

The following rule is adopted.

(I⇒) If $(P \wedge \phi)\{\rho\}Q$ and $(P \wedge \neg\phi) \rightarrow Q$ then $P\{\phi \Rightarrow \rho\}Q$

4.1.4 Disjunction of requirements

To be able to represent all our examples in Section 1, we need to introduce an operator to represent disjunction of requirements. We will use the symbol '+' to represent this. For instance, we might represent the imperative *Open the door or the window!* as follows.

$$\rho_1 = \textit{Open the door} \quad \rho_2 = \textit{Open the window}$$
$$P = \neg(\textit{The window is open} \vee \textit{the door is open}) = (P_1 \wedge P_2)$$
$$Q = \textit{The window is open} \vee \textit{the door is open} = (Q_1 \vee Q_2)$$
$$P\{\rho_1 + \rho_2\}Q = P\{\textit{Open the door} + \textit{Open the window}\}Q$$

If we look at these two requirements separately, they might look like this $P_1\{\rho_1\}Q_1$ and $P_2\{\rho_2\}Q_2$. Typically, a disjunction provides a choice to satisfy one or both requirements that is why it is assumed that both requirements are correct. After a disjunction of requirements is satisfied, we will have that $Q = (Q_1 \vee Q_2)$ is the case. The following rule is adopted.

(I+) If $P\{\rho_1\}Q$ and $P\{\rho_2\}Q$ then $P\{\rho_1 + \rho_2\}Q$

This provides us with the outline for a set of operators for requirements, which allows us to represent all the examples in Section 1.

4.2 Axiomatisation

Using Hoare triples and rules to represent imperatives, we propose the the following axioms and rules. Following Gries (1983) and Harel (1979), we use the notational equivalence $P\{\rho\}Q \equiv P \rightarrow [\rho]Q$.

4.2.1 Definition of sets and terms

Let $C = \{c_1, c_2, \ldots\}$ be a set of constant symbols; $V = \{v_1, v_2, \ldots\}$ be a set of variable symbols; and $F = \{f_1, f_2, \ldots\}$ be a set of function symbols. Further more, let $AtImp = \{r_1, r_2, \ldots\}$ be a set of n-ary atomic actions, and $AtPred = \{p, q, \ldots\}$ be a set of n-ary atomic predicate symbols.

Terms t are defined by

$$t ::= c \mid v \mid f(t_1, t_2, \ldots, t_n)$$

where $c \in C, v \in V, f \in F$, and f is a function in n arguments, and $t_1, t_2, \ldots, t_n$ are themselves terms.

4.2.2 Syntax of imperatives

Imperatives prescribe actions and we have seen that their properties are captured by dynamic operators. Thus, the syntax of the requirements conveyed by imperatives can be defined just as a subset of actions as defined by Harel (1979). If $r \in AtImp$, and ϕ is a formula, then the requirements ρ are defined as follows.

$$\rho ::= r(t_1, t_2, \ldots, t_n) \mid \phi? \mid \rho_1; \rho_2 \mid \rho_1 + \rho_2$$

Here we will represent $\phi \Rightarrow \rho$ by $\phi?; \rho$.

In order to combine statements and requirements, we can define the well formed formulae ϕ of language L as follows. If $p \in AtPred$, $t_1, t_2, \ldots, t_n$ are terms, x is a variable and ρ is an action, then:

$$\phi ::= p(t_1, t_2, \ldots t_n) \mid t_1 = t_2 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \exists x\phi \mid [\rho]\phi$$

We assume the usual classical definitions for $\vee, \rightarrow, \leftrightarrow, \forall$, together with

$$\langle\rho\rangle\phi =_{def} \neg[\rho]\neg\phi$$

4.2.3 Axioms

A0) All tautologies in predicate calculus
A1) $[\phi?; \rho]\psi \leftrightarrow \phi \rightarrow [\rho]\psi$
A2) $[\rho_1; \rho_2]\phi \leftrightarrow [\rho_1]([\rho_2]\phi)$
A3) $[\rho_1 + \rho_2]\phi \leftrightarrow [\rho_1]\phi \vee [\rho_2]\phi$
A4) $[\phi?]\psi \leftrightarrow \phi \rightarrow \psi$
A5) $\forall x(\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \forall x\psi)$ Provided that x is not free in ϕ.
A6) $\forall x\phi(x) \rightarrow \phi(t)$ Provided that t is free in ϕ.

Axiom (A1) combines requirements with statements. ϕ is statement (single or composed) although it is part of an imperative. The axiom expresses that, once we have identified the pre and post-conditions of the conditional imperative ($\phi?\rho$), we can extract ϕ.

4.2.4 Rules

(a) *Modus Ponens* (MP): $\phi, \phi \rightarrow \psi \vdash \psi$
(b) *Necessitation* (Nec): $\phi \vdash [\rho]\phi$
(c) *Universal generalisation* (UG): $\phi \vdash \forall x \phi$ provided x is not free in ϕ.

The following rules can be derived. They include the Hoare rules we use to model properties of imperatives.

(I;)	If $P\{\rho_1\}Q$ and $Q\{\rho_2\}R$ then $P\{\rho_1; \rho_2\}R$.
(I+)	If $P\{\rho_1\}Q$ and $P\{\rho_2\}Q$ then $P\{\rho_1 + \rho_2\}Q$.
(I⇒)	If $(P \wedge \phi)\{\rho\}Q$ and $P \wedge \neg\phi\{\rightarrow\}Q$ then $P\{\phi \Rightarrow \rho\}Q$.
(WP)	If $P\{\rho\}Q$ and $Q \rightarrow \gamma$ then $P\{\rho\}\gamma$.
(SP)	If $\gamma \rightarrow P$ and $P\{\rho\}Q$ then $\gamma\{\rho\}Q$.
(CR)	If $P' \rightarrow P$ and $P\{\rho\}Q$ and $Q \rightarrow Q'$ then $P'\{\rho\}Q'$.
(CDR1)	If $P\{\rho\}Q$ and $P'\{\rho\}Q'$ then $(P \wedge P')\{\rho\}(Q \wedge Q')$.
(CDR2)	If $P\{\rho\}Q$ and $P'\{\rho\}Q'$ then $(P \vee P')\{\rho\}(Q \vee Q')$.

4.2.5 Interpretation

The interpretation for L follows from Harel's (1979) definition of semantics for first order dynamic logic. We only point out here that in Hoare's axiomatisation a state was a computer state. For imperatives a state is a state of affairs as they are perceived by speakers and hearers. The full details of the model and a proof of soundess are presented elsewhere (Pérez-Ramírez 2002).

In brief, actions ρ define a set of relationships between states $k_1, \ldots, k_n$. A state k satisfies a formula ϕ iff $k \models \phi$.

4.3 Encapsulation

The paradoxes described in Section 1.1 arise because the logic of the standard solution leads to counterintuitive conclusions. Hoare logic seems better at capturing some of the properties of imperative.

The way in which Hoare logic is used also provides a distinction between prescribed and derived requirements. Given a program, Hoare logic provides the means to verify whether or not this program is correct. However the logic does not infer the existence of another program or eliminate instructions. A program simply encapsulates a set of instructions. We follow the same idea of encapsulation in the following definitions.

Definition 3 (Set of requirements) *Let $\sigma_k = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$ be a sequence of requirements demanded in context k, such that $\rho_1, \rho_2, \ldots, \rho_n$ represent actions prescribed by imperatives sentences.*

Note that σ_k allows us to distinguish between demanded and derived actions. There is the implicit assumption that all the requirements in σ_k are to be satisfied provided that σ_k is *correct*.

Definition 4 (Correctness of a set of requirements) *A sequence* σ_k *is* correct *with respect to context* k *iff* $k \models P\{\rho_1; \rho_2; \ldots; \rho_n\}Q$ *for appropriate pre/post-conditions* P *and* Q*.*

4.4 Paradoxes

Here we review the paradoxes of Section 1.1 within the proposed formal framework.

4.4.1 Ross' counterexample

Assuming that $\sigma_k = \langle \textit{Post the letter} \rangle$ is requested, we assume the components are instantiated as follows:

$$P = \neg \textit{The letter is posted} \quad \rho = \textit{Post the letter} \quad Q = \textit{The letter is posted}$$
$$P\{\rho\}Q = \neg(\textit{The letter is posted})\{\textit{Post the letter}\}(\textit{The letter is posted})$$

Assuming that this imperative is correct (that is $k \models P\{\rho\}Q$), then there is no way of introducing a disjunction of requirements (represented by the operator '+') in a way that would allow us to infer that a disjunction of imperatives is being requested. That is, we cannot infer $\sigma_k = \langle \rho + \rho' \rangle$ for some other action ρ'.

4.4.2 Jorgensen's dilemma

In Jorgensen's dilemma

(a) *Love your neighbours as yourself!*
(b) *Alison is your neighbour*
___
(c) *Love Alison as yourself!*

if we assume that the order in (a) has been given then we can encapsulate the request as $\sigma_k = \langle \textit{Love your neighbours as yourself!} \rangle$. Adopting free usage of quantifiers, let us define the components as follows.

$$\begin{aligned} P(x) &= x \textit{ is your neighbour} \\ \rho(x) &= \textit{Love } x \textit{ as yourself} \\ Q(x) &= \textit{You love } x \textit{ as yourself} \end{aligned}$$

We can represent (a) as $\forall x(P(x)\{\rho(x)\}Q(x))$. For the purposes of the proof, we will re-express this as

$$\forall x(P(x) \rightarrow [\rho(x)]Q(x))$$

If we assume that (a) and (b) hold, then we can infer that $[\rho(Alison)]Q(x)$, as illustrated below.

(1) $P(Alison)$	assumption
(2) $\forall x(\phi(x) \rightarrow [\rho(x)]Q(x))$	assumption
(3) $\phi(Alison) \rightarrow [\rho(Alison)]Q(Alison)$	(2), universal instantiation
(4) $[\rho(Alison)]Q(Alison)$	1), 3) and MP

This accounts for imperatives and statements participating in inference. The requirement $\rho(Alison)$ in the conclusion is a *derived action* but we are not inferring that it was requested explicitly.

4.4.3 Elimination for conjunction

If the order *Write a letter and send it!*[6] is given, it can be encapsulated as $\sigma_k = \langle Write\ a\ letter; Send\ it \rangle$. In L it is represented by

$$\neg(There\ is\ a\ letter)\{Write\ a\ letter; Send\ it\}(The\ letter\ is\ sent)$$

(as in Section 4.1.2), there is no way of eliminating one of the requirements from the sequence σ_k.

5 Conclusions

We have presented an axiomatisation for imperatives which provides a satisfactory account for some inferential aspects such as Jorgensen's dilemma without being affected by paradoxes such Ross' counterexample.

It is assumed that by various means (order, advice, request, etc.) imperatives convey *requirements*. The analysis of the properties of imperatives allows us to formalise the behaviour of the connectives between imperatives. This behaviour is similar but not identically to that of the connectives in classical logic.

Hoare logic allows us to distinguish between *uttered* and *derived* requirements. Hoare assumes that given a program, his logic provides the means to verify whether the program is correct. In so doing, the logic does not allow us to infer the existence of some other program. In the same way, the axiomatisation presented here can 'verify' imperatives or sequences of imperatives but is not able to infer a new utterance. The explicit distinction between derived and uttered requirements helps to avoid Ross' counterexample.

Propositions and imperatives interact within the axiomatisation. It allows us to verify the appropriate use of imperatives (correctness) where subjective properties play a role. Verification of correctness provides a 'procedure of legitimation' for imperatives. It is able to account for the notions of *satisfiability*, conceived here as the possibility of reaching the state of affairs demanded (or determining that that state of affairs has already been reached).

[6]In this example "*it*" refers to the letter; we are assuming that anaphora are resolved.

The axiomatisation presented here is only a first step towards a model for imperatives. An analysis of imperatives has applications not only in NLP but also in other fields such as agents (Vere and Bickmore, 1990; Piwek, 2000; 2001) information management (Martino, 1981), law etc.

Extensions for generalised quantifiers have been proposed (Pérez-Ramírez 2002). Treatment of other problems such as satisfaction, agency and time, and "contrary to duty" imperatives (Prakken and Sergot, 1996; Alarcón-Cabrera, 1998) would be good extensions. Further work includes an exploration of the relationship with dynamic logic as used in the analysis of propositional discourse (Groenendijk and Stockhof, 1991; van Eijck and Vries, 1992). Work in implementing such dynamic theories (van Eijck, 1998, for example) may also be applicable.

References

Alarcón Cabrera Carlos, 1998. "*Von Wright's Deontic Logics and 'Contrary-to-Duty Imperatives.'*" Ratio Juris. Vol 11. No. 1 March 1998. 67- 79.

Alchourrón Carlos, Martino Antonio, 1990. "*Logic Without Truth.*" Ratio Juris. March 1990. 46-67.

Buvac, Sasa. 1995. "*Resolving Lexical Ambiguity Using a Formal Theory of Context.*" Manuscript, Stanford. `http://www-formal.Stanford.EDU/buvac/`.

Chellas, B., 1971. "*Imperatives.*" Theoria. Vol 37, 114-129. 1971

Castilho et al, Marcos A. Olivier Gasquet, Andreas Herzig: "*Formalizing Action and Change in Modal.*

Logic I: the frame problem." Journal of Logic and Computation Volume 9, Issue 5: October 1999. 701-735.

De Roeck A.; Ball, R.; Brown, E.K.; Fox, C.J.; Groefsema, M.; Obeid, N.; and Turner, R. 1991. "*Helpful Answers to Modal and Hypothetical Questions.*" In Proceedings of the European ACL 1991, Berlin. 257-262.

Fulda S., Joseph, 1995. "*Reasoning with Imperatives Using Classical Logic.*" SORITES ISSN 1135-1349. Issue No. 3. November 1995. 7-11.

Gries, David, 1983. "*The Science of programming.*" Department of Computer Science. Cornell University. Upson Hall Ithaca, NY. 1983.

Groenendijk and Stokhof, 1991. "*Dynamic predicate logic.*" Linguistic and philosophy, 14:39-100.

Hamblin, C. L, 1987. "*Imperatives.*" Basil Blackwell. USA. 1987

Hare R. M., 1961. "*The Language of Morals.*" Oxford at the Clarendon Press. Reprinted in 1961.

Harel David, 1979. "*First-Order Dynamic Logic.*" Lecture Notes in Computer Science. Edited by Goos and Hartmanis. 68. Springer-Verlag.Yorktown Heights, NY. 1979.

Hoare. C. A. R., 1969. "*An Axiomatic Basis for Computer Programming.*" Communications of the ACM, Vol. 12, No 10. October 1969. pp. 576 -580, 583.

Huntley Martin, 1984. "*The Semantics of English Imperatives.*" Linguistics and Philosophy. Vol 7. 1984. 103-133.

Jorgensen Jorgen, 1937. "*Imperatives and logic.*" Erkenntnis. Vol. 7, (1937-1938), pp. 288-296.

Kamp, H., 1973. "*Free-Choice Permission.*" Chicago: University of Chicago Press. Kamp, H. 1973. Proc. of the Aristotelian Society, N.S. 74: 57-74.

Levinson Stephen C., 1983. "*Pragmatics.*" Cambridge textbooks in linguistics. Cambridge University Press. 1983.

MacFadyen Heather, 1996. "*Using Verb Moods.*" HyperGrammar. University of Ottawa. 1996.

Manara and De Roeck 1996. "*A Belief-Centered Treatement of Pragmatic Presuppositions*" LACL 1996: 274–291.

Manara and De Roeck, 1997. "*Context as Partial Beliefs, and the Pragmatic Modelling of Presuppositions.*" Context 97. Brazil. 66-74.

Martino A. Antonio, 1981. "*Deontic logic, Computational linguistics and legal information systems.*" Proceedings of the first international conference on logic, informatics, law. Vol. 2. Florence, Apr. 6-10. 1981.

Megginson David, 1996. "*The Purpose of a Sentence.*" HyperGrammar. University of Ottawa. 1996.

Meyden R. van der, 1996. "*The Dynamic Logic of Permission.*" Journal of Logic and Computation, Vol 6, No. 3 pp. 465-479, 1996. A version of this paper appeared at the IEEE Symposium on Logic in Computer Science, Philadelphia, 1990.

Meyer J.-J.Ch., Dignum F., and Wieringa R.J., 1996. *"Free choice and contextually permitted actions."* Studia Logica, 57:193-220, 1996.

Norman, Timothy. J. and Reed, Chris, 2000. "*Delegation and responsibility.*" Intelligent Agents VII: Proceedings of the Seventh International Workshop on Agent Theories, Architectures and Languages. Boston, MA, July 2000.

Pérez-Ramírez, M., 2000. "*Imperatives, state of the art.*" CLUK 3. University of Brighton. UK. April, 2000.

Pérez-Ramírez, M., 2002. "*Formal pragmatic model for imperatives interpretation*" University of Essex. UK. (draft thesis).

Piwek, P., 2000. "*Imperatives, Commitment and Action: Towards a Constraint-based Model.*" In: LDV Forum: Journal for Computational Linguistics and Language Technology, Special Issue on Communicating Agents, 2000.

Piwek, P., 2001. "*Relating Imperatives to Action.*" In: Bunt, H. and R.J. Beun, Cooperative Multimodal Communication, Lecture Notes in Artificial Intelligence Series 2155, Springer, Berlin/Heidelberg. 2001.

Prakken, Henry and Sergot, Marek, 1996. "*Contrary-to-duty Obligations.*" Studia Logica 57(1/2):91-115 (1996)

Radford Andrew, 1997. "*Syntactic theory and the structure of English. A minimalist approach.*" Cambridge University Press. 1997.

Ross A., 1941. "*Imperatives and Logic.*" Theoria (journal). Vol. 7. 53-71. 1941.

Segerberg Krister, 1990. "*Validity and Satisfaction in Imperative Logic.*" Notre Dame Journal of Formal Logic. Volume 31, Number 2, Spring 1990. 203-221.

Stirling. 1992. Stirling, Colin. "*Modal and temporal logics.*" Handbook of Logic in Computer Science, vol. 2. Edit. S. Abramsky and D. Gabbay and T. Maibaum. Publisher, Oxford University Press. 477-563. 1992

Tarski, Alfred, 1944. "*The semantic conception of truth and the foundations of semantics.*" University of California, Berkeley. Published in Philosophy and Phenomenological Research 4 (1944).

Vere, Steven and Bickmore, Timothy, 1990. "*A Basic Agent.*" Computational Intelligence, Vol. 6,

van Eijck and Vries, 1992. "*Dynamic interpretation and Hoare deduction."* Journal of Logic, Language and Information, 1(1):1-44, 1992.

van Eijck, 1998b. "*Dynamo - A Language for Dynamic Logic Programming.*" Manuscript, CWI/ILLC, December 1998.

von Wright George Henrik, 1968. "*An Essay in Deontic Logic and The General Theory of Action.*" North Holland Publishing Company-Amsterdam. 1968.

Walter Robert, 1996. "*Jorgensen's Dilema and How to Face It.*" Ratio Juris. Vol 9. No. 2 June 1996. 168-71.

Weinberger Ota, 1991. "*The Logic of Norms Founded on Descriptive Language.*" Ratio Juris. 1991. 284-307.