Technical Report No. 2009-554

An Exploration of Semantic Formalisms - Part I: A Comparison of First Order Predicate Calculus, Intensional Logic and Conceptual Structures*

Craig Thomas
School of Computing, Queen's University
Kingston, Ontario, Canada K7L 3N6

February 10, 2009

Abstract

This summary paper describes the general properties and features of two systems of semantic representation: Montague's Intensional Logic and Jackendoff's Conceptual Structures. Each system of representation is based upon a different type of semantic theory, and thus each one is meant to express different semantic and linguistic phenomena. The basic concepts of how each formalism works and their expressiveness will be explored. Several linguistic phenomena will be explained and used as test cases to demonstrate the expressiveness and limitations of each semantic formalism. Additionally, the first order predicate calculus is explained and used to express linguistic phenomena as an aid for the reader.

Contents

*This research was conducted with funding assistance from the Natural Sciences and Engineering Research Council of Canada.

1 Introduction

Research in the areas of formal semantics and computer science has resulted in a new and exciting area called *computational semantics*. This important branch of computational linguistics is concerned with the *automated processing* of meaning obtained from natural language expressions (Blackburn and Bos, 2003). Paramount to the field is the development of *semantic formalisms* that are suitable for automated processing (Blackburn and Bos, 2003). The purpose of these semantic formalisms is to systematically, precisely and unambiguously describe meaning underlying natural language expressions. Semantic formalisms have applications in areas such as machine translation (Lønning and Oepen, 2006), natural language generation (Donald, 2006), information storage and retrieval (Koyama *et al.*, 1998), dialogue systems (Traat and Bos, 2004) and information extraction (Bollegala *et al.*, 2007).

The expressiveness of a semantic formalism may be evaluated based on a number of criteria including linguistic coverage, precision and accuracy. First, a formalism must have appropriately wide *linguistic coverage* in the sense that it must be possible to express a large number of different linguistic phenomena such as quantification, tense, modality and propositional attitudes (Bos, 2005; Donald, 2006). Coverage may also be calculated with respect to a set of natural language *domains* the formalism is capable of operating within. Such domains may include airline ticket purchases, weather reports, sports announcements and lunar landing scenarios (Ammicht *et al.*, 2007; Donald, 2006; Weischedel, 2006). Second, the formalism must allow for the *precise* expression of semantic information. For example, the formalism must be able to express the differences between the English expressions "few" and "a few". Third, the formalism must be *accurate* with respect to the meaning it attempts to describe. For example, the formalism must be able to describe exactly and correctly what the quantification "few" means. Other evaluation criteria may include human-usability aspects and coverage over semantic phenomena such as entailment, synonymy, polysemy and antonymy (Donald, 2006; Davis and Gillon, 2004).

Formalisms operate in different ways depending on the type of *semantic theory* they are based upon. The classification of semantic theories into meaningful categories is complex, as the study of semantics has attracted researchers from a number of diverse disciplines including philosophy, logic, psychology and linguistics (Davis and Gillon, 2004). Further complications arise as many theories attempt to account for overlapping sets of semantic phenomena, thus making it difficult to draw clear and insightful distinctions. Following from Davis and Gillon (2004), the classification approach used in this paper will divide semantic theories into two categories: reference theories and meaning theories.

1.1 Reference Theories

Reference theories are concerned with the *denotation* of linguistic expressions relative to a single reference coordinate. This denotation, typically referred to as an *extension*, comes in different forms depending on the type of expression. To illustrate this point, consider the expression "the Emperor of Rome". This statement reflects a set of individuals, one for each possible world and time. The extension of this expression is an individual at a single world and time coordinate. For "Caesar is the Emperor of Rome", the extension is a *truth value* obtained by evaluating a set of predicates at a particular time and world coordinate (Dowty, 1981).

Reference based formalisms evaluate the extension of an expression to relative *reference coordinates*. Lewis (1972) refers to these coordinates as *contextual coordinates* which may include time, possible world, speaker, audience and place. Formalisms based on reference theories are typically realized as model-theoretic constructs based on truth semantics and possible worlds[1]. One popular reference theoretic based formalism is the first order predicate calculus. In unextended versions of the first order predicate calculus model, time, possible world, speaker or any other contextual coordinate type are unavailable. Thus all extensions in this language are relative to a single reference point. Variations of the first order predicate calculus (e.g. first order tensed logic, first order modal logic) have been developed that integrate additional coordinate reference sets (Saeed, 2003).

Extensions only express a narrow semantic value of a statement (Dowty, 1981). Consider the sentence "Caesar seeks a conspirator". One possible extension is evaluated based on truth-conditional statements expressing that Caesar is seeking a conspirator that we *know* to be Brutus. This particular extension can be captured with the first order predicate calculus. The wider set of semantic values, known as the *sense* or *intension* of the expression, are calculated relative to other sets of referential coordinates. For example, another "sense" of "Caesar seeks a conspirator" is evaluated based on truth-conditional statements expressing that Caesar is seeking a conspirator despite the fact that none exist. This sense can only be obtained by evaluating the expression relative to a set of possible worlds that agree with Caesar's belief that a conspirator *actually* exists. In other words, the proper denotation can only be obtained by examining extensions using referential coordinates that differ from the way the world actually is.

1.2 Meaning Theories

Meaning theories are concerned with the overall meaning or *sense* that is expressed by a linguistic construction (Davis and Gillon, 2004). This sense, referred to as an *intension*, contains the set of all possible extensions for an expression (Dowty, 1981). For example, the intension of the expression "the Emperor of Rome" would contain the set of all the possible individuals that the Emperor of Rome may have referred to in the past, present or future. Thus if we consider only the time coordinate, then at time t_1 the denotation may be Augustus while at time t_2 the denotation may be Tiberius. For the expression "Caesar seeks a conspirator", the set of extensions would contain the instance where Caesar is seeking

[1]Possible worlds simply describe different states of affairs. The world we live in, called the *actual world* is described by a series of circumstances. Other *possible worlds* are described by circumstances that differ from our own in infinitely many ways (Kearns, 2000).

Brutus, where Caesar is seeking *any* conspirator, and where Caesar is seeking a conspirator despite the fact that *none exist*. To summarize, with meaning theories the value of an expression refers to the *entire set* of extensions or values that an expression may contain. Formalisms based on meaning theories are typically realized as series of feature structures that correspond to basic, recognizable concepts.

Meaning theories are further sub-categorized when we examine how we assign sense to lexical items. Consider the lexical item "assassin". An *atomic* meaning theory would stipulate that "assassin" is nothing more than a structureless concept [assassin] that cannot be decomposed (Davis and Gillon, 2004). Stated another way, there is no *meaning relation* between "assassin" and "a murderer who kills for monetary gain", since each expression has an isolated meaning. Opposed to atomic meaning theories are *molecular* meaning theories that decompose simple expressions into other meaning constituents (Wierzbicka, 1996). These types of meaning theories can further be subdivided into the classifications of *analytic* and *encyclopedic*.

Analytic molecular meaning theories distinguish between two types of sentences (Davis and Gillon, 2004). Consider the sentence "assassins are murderers who kill for monetary gain". This type of sentence is *analytic* in nature as it defines a word in terms of other words, whereas the sentence "human blood is red" is *synthetic* since it talks about the nature of the world. Both types of sentences allow for meaning *entailments*. For example, unlike atomic meaning theories, "assassin" would carry a meaning relation to "a murderer who kills for monetary gain". *Encyclopedic* molecular meaning theories do not distinguish between synthetic and analytic sentence types (Davis and Gillon, 2004). As a result, word meaning and world nature become fused in the lexical definition of a word. To demonstrate, consider the words "conspirator" and "assassin". If an individual Caesar holds a belief that conspirators are assassins, then "assassin" will be in the lexical definition of "conspirator". Thus Caesar will be able to infer "Brutus is an assassin" directly from "Brutus is a conspirator".

1.3 Goal of the Paper

The purpose of this paper is to examine the expressiveness of semantic representations of two different types, each one based upon a different semantic theory. Specifically, this paper will explore the semantic formalism developed by Montague (1974a) known as Intensional Logic, and the formalism developed by Jackendoff (1990) known as Conceptual Structures. To explain the basic machinery of each, a series of simple linguistic phenomena in English will be presented and expressed using both formalisms[2]. Additionally, first order predicate calculus representations will be provided and explained for each linguistic example. This is done as an aid for the reader and to demonstrate how each formalism attempts to correct for problems or insufficiencies with the traditional first order predicate calculus approach to express meaning. The expressiveness of each formalism will be evaluated with respect to the precision, accuracy and human-usability of their resultant semantic statements. By virtue of the fact that our comparison involves a variety of linguistic phenomena, the coverage of each formalism will also be examined.

The remainder of this paper will proceed as follows. Section 2 will explain the basic machinery underlying the first order predicate calculus, intensional logic and conceptual

[2]English sentence fragments will be used exclusively, since the intended audience is expected to be well versed in the English language. The reader is reminded that any natural language could be used in place of English, as the underlying meaning of a linguistic expression is intended to be language independent in nature.

structures using a series of simple linguistic concepts. Section 3 will compare the expressive power behind each formalism by exploring how each one successfully (or inadequately) captures the meaning of more advanced linguistic concepts. Finally, section 4 will summarize the differences between formalisms and provide some concluding remarks.

2 A Tale of Three Formalisms

The purpose of this section is to introduce the basic machinery of each formalism. To accomplish this task, simple English expressions will be provided, with their semantic equivalents expressed using the first order predicate calculus, intensional logic and conceptual structures. The intended result is to provide the reader with enough background explanation of each formalism to understand their operation in relation to the more complex linguistic examples which will be explored in section 3. The principles behind each model, as well as their basis in semantic theory are described briefly below.

The first order predicate calculus has been used extensively as a formalism for capturing the semantics of natural languages, despite the fact that only a very loose methodology exists for translating the meaning of a natural language expression into its first order logical equivalent. The attraction of this particular formalism comes from the extensive set of computational tools for various inferencing and automated reasoning tasks (Blackburn and Bos, 2003). The first order predicate calculus is a reference theoretic formalism that is based upon a model M that consists of the tuple $\langle A, F\rangle$ where A is a domain of individuals (the set of entities e) and F is a function that assigns values to constant terms (names and predicates) (Dowty, 1981). Predicates are functions that take zero or more terms as arguments and return truth values. Additionally, it is possible to assign values to variable terms through the use of a variable assignment g. The truth value of a particular expression is always obtained relative to the model M and any variable assignments g (Dowty, 1981). The semantics of the operators $\wedge$, $\vee$, $\leftrightarrow$, $\rightarrow$, $\forall$ and $\exists$ will not be covered here as computer scientists are assumed to be familiar with their use and definition.

Intensional logic is a reference theoretic formalism that extends the model that the first order predicate calculus is based upon. Montague (1974a) defined a rigorous methodology for building a grammar by requiring that each syntactic formation rule be coupled with a well defined semantic interpretation. Montague realized that complex syntactic constituents are built from relatively simple ones. Thus his treatment of a grammar ensures that every syntactic formation rule has a well defined semantic interpretation, which in turn leads to well defined semantic definitions for any production in the language (Partee, 1972; Montague, 1974a). Semantic rules are specified using intensional logic, a formalism based on a model M that consists of the quintuple $\langle A, W, T, <, F\rangle$ where A is a domain of individuals, W is a domain of possible worlds, T is the domain of possible times, $<$ is an ordering on T, and F is a function that assigns values to constant terms relative to a (time, world) pair (Dowty, 1981). Variable assignments g also exists in the model, again used to resolve values for bound variables. Since it is type theoretic in nature, the formalism contains two basic types: entities (e) and truth values (t). Additional types are defined recursively, with a derived *sense* type (Dowty, 1981). In addition to the logical and quantification symbols of the first order predicate calculus, Montague defines operators $\hat{}$, $\check{}$, F, P, =, $\Box$ and $\diamond$ which will be discussed throughout the paper.

Finally, conceptual structures are based on an encyclopedic molecular meaning theory. The model that Jackendoff (1990) describes involves the notion of feature structures that express basic concepts. These *conceptual structures* form the fundamental building blocks that humans are assumed to use to build a mental representation of linguistic and non-linguistic information (Jackendoff, 1983). The basic unit of information contained within a conceptual structure is a *concept*. Jackendoff (1990) suggests that every human has an innate set of *conceptual primitives* that can be combined with learned information using *conceptual formation rules* to provide a rich description of possible meanings (Jackendoff, 1983; Wierzbicka, 1996). Each major content-bearing syntactic structure in an expression has a mapping to a well defined conceptual structure. The overall meaning of a sentence is carried by combinations of various structures obtained from the syntax of the expression.

2.1 Names and Individuals

To begin our examination, consider the name "Brutus" which relates to the semantic individual Marcus Junius Brutus. The name is represented below using first order predicate calculus, intensional logic and conceptual structures:

$$\text{brutus} \tag{1}$$

$$\text{brutus'} \tag{2}$$

$$\left[_{Thing}\ \text{BRUTUS}\ \right] \tag{3}$$

Although the representation in (1) does not constitute a well-formed first order predicate calculus formula, it demonstrates how syntactic names are represented by individual constants[3] (Korfhage, 1966). The actual semantic value of the constant term brutus maps to the individual Marcus Junius Brutus in our model. Other constant terms such as "marcus" and "assassin" could also map to the same semantic value. The notation $[\![\alpha]\!]^{M,g}$ will be used when we talk about the denotation of the term α relative to the model M and variable assignment g (Dowty, 1981). Thus $[\![\text{Brutus}]\!]^{M,g}$ = Marcus Junius Brutus.

The intensional logic representation in (2) bears a strong resemblance to the first order predicate calculus representation in (1). The constant term brutus' functions in much the same way as it did in (1), except that brutus' refers to a specific, numbered constant term. There are an infinite number of constant and variable terms available in intensional logic - constants denoted $C_{n,<a>}$ (the set of constants of type a) and variables denoted $V_{n,<a>}$ (the set of variables of type a) (Dowty, 1981). Primed terms are used as convenient labels for otherwise complicated variable names. For example, brutus' may actually be constant $C_{46,e}$, the constant term number 46 of the type entity. A variable x is simply a convenient label for what may be variable term 156 of the type entity, denoted $V_{156,e}$. The primed notation and other "English-like" labels simply allow for a more natural way of referring to various terms. Moving on to semantic values, the denotation of a term is obtained slightly differently than the first order predicate calculus equivalent. Where the first order predicate calculus extension was evaluated relative to the model M and the variable assignment g, the intensional logic extension is obtained relative to M and g as well as a time coordinate and world coordinate (Dowty, 1981). Thus $[\![\text{brutus}]\!]^{M,w,t,g}$ = Marcus Junius Brutus. Unless

[3] One should note that the first order predicate calculus representations used throughout this paper break from traditional notation. Multi-letter constant terms such as "brutus" and "caesar" will be used instead of traditional single letter constant terms "b" and "c" (Korfhage, 1966; Rubin, 1990). Multi-letter predicates are also employed, again breaking notational conventions but, it is hoped, improving clarity.

otherwise stated, the w and t coordinate (chosen from the sets W and T respectively) are considered to be reflective of the actual world and the current time[4]. Finally, representation (2) does not reflect a well formed formula. A correct way of expressing the term "brutus" would be to express it in relation to the set of properties that define it. We will explore properties of individuals in the next section.

Turning to Jackendoff's formalism, representation (3) contains a *conceptual structure*. As stated before, conceptual structures are representations of mental constructs used by humans. A conceptual structure may have one or more conceptual constituents, each with a specific *type*. The set of types is defined as *Thing, Event, State, Action, Place, Path, Property, Manner* and *Amount*[5] (Jackendoff, 1983, 1990). Representation (3) expresses a conceptual structure that is composed of a *Thing* BRUTUS that is understood to be Marcus Junius Brutus.

2.2 Properties of Individuals

In some instances, it is advantageous to talk about the set of properties that define an individual. For example, the set of properties that may define Brutus would be: male, animate and conspirator[6]. With the first order predicate calculus, intensional logic and conceptual structures, the following representations would suffice:

$$\text{MALE(brutus)} \wedge \text{ANIMATE(brutus)} \wedge \text{CONSPIRATOR(brutus)} \tag{4}$$

$$\lambda \text{P}[\check{\ }\text{P(brutus')}] \tag{5}$$

$$\left[_{Thing} \text{ BRUTUS } \right] \tag{6}$$

With the first order predicate calculus, there is no way refer abstractly to the set of properties that define a term. Since properties are expressed using predicates, the only way to talk about the set of properties that defines Brutus would be to create a conjunctive list of all the applicable one-place predicates, which we see in representation (4). Predicates and their denotations are discussed in more detail in the next section.

The intensional logic expression in (5) introduces the λ operator and the extension operator ˇ. The λ-operator is a *function abstractor*, and in this context is used to make an abstract reference to the characteristic functions (P) that describe the properties of "brutus" (see section 2.3 for the definition of a characteristic function) (Dowty, 1981). For example, when combined with the predicate male', the expression λP[ˇP(brutus')](male') would reduce using λ-conversion to ˇmale'(brutus'), which effectively is a truth-conditional statement that says that Brutus is male. The λ-operator can be used to abstract over any type in the language. Thus an abstraction over terms, predicates, relations between predicates or any other complex type is allowed.

The extensional operator ˇ attached to the term P in (5) has a profound effect on the denotation. The ˇ forces the value of the abstract predicate P to be read relative to a world, time pair of coordinates we are using while interpreting the model (Dowty, 1981). Inversely, the intensional operator ˆ would be used if we wanted to refer to the *sense* or

[4]Montague uses I and J in favour of W and T (Dowty, 1981).

[5]This list is not inclusive of all the possible conceptual types. It is based on assumptions about how conceptual "parts of speech" may be organized (Jackendoff, 1990).

[6]This list is not exhaustive, but rather representative of some of the properties an individual may exhibit.

entire set of characteristic functions that P could potentially be across all world and time coordinates (Dowty, 1981). For example, consider the predicate “the Emperor of Rome”. The truth value of the expression $\exists x[{}^{\vee}\text{emperor\_of\_rome'}(x)]$ would be true if there was some individual x who is the Emperor of Rome at a given world, time pair coordinate. In contrast, the expression $\exists x[{}^{\wedge}\text{emperor\_of\_rome'}(x)]$ would be true if there were some individual x who is the Emperor of Rome at *any* point in time and in *any* possible world. The ˇ operator cancels the intensional operator, thus the expressions $^{\wedge\vee}\text{male'}(x)$ and male'(x) are equivalent, being extensions of the male' predicate at the current time and world coordinate (Dowty, 1981). Any predicate or designator without a ˇor ˆoperator is understood to be read only as an extension. Examining this notion from the perspective of types, ˇemperor_of_rome is of the type <e, t>, which is a function (emperor_of_rome at the given world, time coordinate) that takes an entity and returns a truth value. The type of ˆemperor_of_rome is <s, <e, t>>, which is the set of functions (emperor_of_rome at every world, time coordinate) that takes an entity and returns a truth function (Dowty, 1981).

Turning to conceptual structures, the constituent BRUTUS introduced in (6) is implicitly understood to contain the set of properties that define Marcus Junius Brutus. Where first order predicate calculus terms simply refers to a real world object, the structure in (6) is a fully realized conceptual structure of the object in question. It is possible to pick out a property associated with a constituent and explicitly make mention of it for syntactic realization. For example, knowing that Brutus is a man, the following conceptual structure would be equivalent to (6):

$$\left[_{Thing}\ \begin{array}{c} \text{BRUTUS} \\ [_{Property}\ \text{MALE}\ ] \end{array}\right] \tag{7}$$

With representation (7), *restrictive modification* is used to modify the *Thing* BRUTUS and indicate that Brutus is a MALE. The *Property* MALE becomes *fused* with the conceptual constituent *Thing* BRUTUS. This notation usually only serves to explicitly mark features of a conceptual constituent that are important for syntactic purposes. For example, restrictive modification would be employed in instances where adjectives modify nouns (e.g. *red* knife) (Jackendoff, 1983, 1990).

It is also worthwhile to note that the *fusion* of the *Property* MALE with the *Thing* BRUTUS results in a *Thing* that maintains all the same semantic properties that it once did, but without any duplication of properties that the fusion may have introduced (Jackendoff, 1990). To understand this concept, consider representation (7) where there is an implicit *Property* MALE that is already associated with the *Thing* BRUTUS. After the fusion of the *Property* MALE with the *Thing* BRUTUS, the resultant structure would have two MALE *Properties*. One property is the *implicitly* associated property, the other is the *explicitly* marked property. This double specification has absolutely no effect on the conceptual meaning, and therefore we *delete* one of the MALE *Properties* from the structure. In this instance, the explicitly marked MALE *Property* remains. In instances where we fuse two explicitly marked properties, the result is a structure where only one will remain.

2.3 Predicates

Predicates are simply relationships between various entities (Kearns, 2000). In the previous section, we abstractly referred to a series of predicates which described certain properties of individuals. Consider the sentence "Brutus is an honourable man" represented below[7]:

$$\text{HONOURABLE(brutus)} \wedge \text{MALE(brutus)} \tag{8}$$

$$\check{}\text{honourable'(brutus')} \wedge \check{}\text{male'(brutus')} \tag{9}$$

$$\left[\begin{array}{c} \text{BE}_{Ident} \left(\begin{array}{l} \left[\begin{array}{c} \text{BRUTUS} \\ {}_{Thing} \quad [_{Property}\ \text{MAN}] \end{array} \right], \\ [_{Place}\ \text{AT}_{Ident}([_{Property}\ \text{HONOURABLE}\ ])] \end{array} \right) \\ {}_{State} \end{array} \right] \tag{10}$$

The first order predicate calculus representation in (8) contains a series of predicates that we have already seen, joined together by the logical operator $\wedge$. If read literally, this statement expresses that "Brutus is honourable and Brutus is a man". Unfortunately, the logical operator $\wedge$ must be employed to convey meaning despite the lack of evidence in the English statement that such a logical operator is necessary[8] (Jackendoff, 1983). Additionally, there is no indication as to whether these properties are transient or immutable. Whereas immutable properties such as genetic traits may be implicitly connected to an individual, a transient property such as *honourable* may depend on an exact time or set of circumstances (Kearns, 2000).

Turning to the predicates themselves, the predicate MALE represents a function f_{MALE} that maps elements from the domain of individuals into the *truth value* or *t* domain defined as {0, 1} (Dowty, 1981). The function f_{MALE} is known as the *characteristic function* of the set MALE, as it defines what members are part of the set MALE, and those that are not. Entities will be mapped to 1 if they are in the set, or 0 if they are not. For example, given our entity domain {Marcus Junius Brutus, Gaius Julius Caesar, Juliet Capulet}, the function f_{MALE} would be {Marcu Junius Brutus $\rightarrow$ 1, Gaius Julius Caesar $\rightarrow$ 1, Juliet Capulet $\rightarrow$ 0}. Similar treatment exists for the predicate HONOURABLE.

The treatment of the intensional logic expression in (9) is much the same as in (8). Again the terms male' and honourable' are understood to represent the characteristic functions that would map entities into the domain {0, 1}. The difference in the Montague model is that the characteristic functions of each predicate may change depending on the time and world coordinate (Dowty, 1981). For example, at time t_1 the characteristic function for honourable' ($f_{honourable'}$) may have the mapping {Marcus Junius Brutus $\rightarrow$ 1, Gaius Julius Caesar $\rightarrow$ 1, Juliet Capulet $\rightarrow$ 1}, but at time t_2 it may be {Marcus Junius Brutus $\rightarrow$ 0, Gaius Julius Caesar $\rightarrow$ 1, Juliet Capulet $\rightarrow$ 1}. This allows us to capture the fact that some properties may be transient, since the resultant values of the function may change depending on the referential coordinates used relative to the model. Since the expression in (9) uses the extensional operator ˇ the truth value of the function honourable' will be relative to the current time and actual world.

Examination of the conceptual structure expressed in (10), reveals the use of a *function-argument structure* (Jackendoff, 1990). Function argument structures take conceptual constituents as arguments and relate them to each other in various ways. The semantic type of the function is defined by the nature of the relationship it expresses. For example, a

[7]For the sake of simplicity, we choose to express the state of being a *man* as a consequence of being male. A more rigorous definition of man would be a combination of *adult* and *male*.

[8]In English, *and* usually translates to the use of $\wedge$. However, *and* is not present in the statement "Brutus was an honourable man".

GO function is an *Event* where the BE function would be a *State*. Furthermore, the set of functions is limited to a set of *conceptual primitives* that cannot be further decomposed into any other sets of functions[9]. If read literally, the representation in (10) reads as a *State* containing the function BE which relates the *Thing* BRUTUS as being AT the *Property* HONOURABLE. Representing this property as a *Place* suggests that an object may move from that location to someplace else, correctly capturing the transient nature of HONOURABLE.

Although this structure may at first appear to be more complex than the first order predicate calculus equivalent, Jackendoff (1983) notes that the subscripts may be dropped once the reader is familiar with the types of each constituent and function-argument structure. Throughout the remainder of this paper, subscripts will be omitted where the types of the semantic objects have already been introduced. This is done to improve clarity and reduce clutter. As an example, (10) may be rewritten as:

$$\left[\ \mathrm{BE}_{Ident}\left(\begin{array}{l}\left[\begin{array}{c}\mathrm{BRUTUS}\\ [\ \mathrm{MAN}\ ]\end{array}\right],\\ [\ \mathrm{AT}_{Ident}([\ \mathrm{HONOURABLE}\ ])]\end{array}\right)\right] \tag{11}$$

2.4 Verb Phrases

Moving to more complex linguistic examples, consider the sentence “Brutus stabbed Caesar”[10]. Both the first order predicate calculus representation and conceptual structure representation are given below:

$$\mathrm{STAB}(\mathrm{brutus},\ \mathrm{caesar}) \tag{12}$$

$$\lambda \mathrm{P}[{}^{\vee}\mathrm{P}(\mathrm{brutus'},\ \mathrm{caesar'})](\mathrm{stab'}) \tag{13}$$

$$\left[\begin{array}{l}\mathrm{CAUSE}\left(\begin{array}{l}[\ \mathrm{BRUTUS}\ ],\\ \left[\ \mathrm{GO}\left(\begin{array}{l}[_{Thing}\qquad],\\ [_{Path}\ \mathrm{TO}([_{Place}\ \mathrm{IN}([\ \mathrm{CAESAR}\ ])])]\end{array}\right)\right]\end{array}\right)\\ [_{Manner}\ \mathrm{PUNCTURE}\ ]\end{array}\right] \tag{14}$$

New complexity is found in the form of extra argument places for the predicate STAB in the first order predicate calculus representation in (12). The first argument place indicates the *agent* performing the action and the second argument place indicates the *patient* of the action. The terms *agent* and *patient* are known as *thematic roles* or *θ-roles*, and are used as informal markers to describe the contents of predicate arguments (Haegeman, 2005). A verb assigns a θ-role to each of its arguments, and a rule known as the *θ-Criterion* states that each syntactic constituent may receive one and only one thematic role (Haegeman, 2005; Poole, 2002; Radford, 1997).

The intensional logic expression in (13) contains a non-reduced representation of the sentence. Although we have seen λ-expressions before in (5), this one differs in that there

[9] As of Jackendoff (1990), the set of conceptual primitives has not been fully enumerated as it is still uncertain which functions should not be further decomposed into other functions.

[10] To simplify the explanation of various linguistic phenomena and their treatment, all the first order predicate calculus, intensional logic and conceptual structure examples will have the default reading of the past tense for now. Section 3.1 will talk more about the difficulties surrounding tense.

is the possibility for it to be reduced to a simpler form. All expressions containing λ-terms in intensional logic should be reduced to their simplest form wherever applicable. Further reduction of (5) was not possible since it was left unspecified what predicate should be combined with the λ-bound term. However, since a predicate is specified in (13), the expression should undergo λ-conversion as demonstrated below:

$$\begin{array}{c}\lambda \mathrm{P}[\check{\ }\mathrm{P}(\mathrm{brutus'},\ \mathrm{caesar'})](\mathrm{stab'}) \\ \downarrow \\ \check{\ }\mathrm{stab'}(\mathrm{brutus'},\ \mathrm{caesar'})\end{array} \tag{15}$$

As we can see, the converted representation in (15) and (12) are completely equivalent statements. This equivalence is made possible due to the fact that we are only examining the *extension* of the predicate in (13).

The conceptual structure representation in (14) literally reads that the *Thing* BRUTUS caused an unspecified *Thing* to be placed inside of CAESAR. The addition of the restrictive modifier PUNCTURE is used to indicate the *Manner* in which the action is performed. Although the complexity of this structure appears to make it much more unwieldy than the first order predicate calculus representation, it is possible to represent this action in another way with a different structure. More details of this simplification are given below.

2.5 Prepositional Phrases

Consider the sentence "Brutus stabbed Caesar with a knife". Although we have a basic form for the STAB action, the prepositional phrase "with a knife" forces the addition of another argument place in the first order predicate calculus representation. An additional worry is the complexity of the conceptual structure representation. However, it is possible to simplify the latter by using different *primitive* functions. Representations for "Brutus stabbed Caesar with a knife" are below:

$$\exists \mathrm{x}(\ \mathrm{KNIFE}(\mathrm{x}) \wedge \mathrm{STAB}(\mathrm{brutus},\ \mathrm{caesar},\ \mathrm{x})\ ) \tag{16}$$

$$\exists \mathrm{x}[\ \check{\ }\mathrm{knife'}(\mathrm{x}) \wedge \check{\ }\mathrm{stab'}(\mathrm{brutus'},\ \mathrm{caesar'},\ \mathrm{x})\ ] \tag{17}$$

$$\left[\begin{array}{l}\mathrm{CAUSE}\left(\begin{array}{l}[\ \mathrm{BRUTUS}\ ], \\ \left[\ \mathrm{GO}\left(\begin{array}{l}[_{Thing}\ \mathrm{KNIFE}\ ], \\ [_{Path}\ \mathrm{TO}([_{Place}\ \mathrm{IN}([\ \mathrm{CAESAR}\ ])])]\end{array}\right)\ \right]\end{array}\right) \\ [_{Manner}\ \mathrm{PUNCTURE}\ ]\end{array}\right] \tag{18}$$

Unfortunately, the first order predicate calculus representation in (16) strays away from the English it attempts to describe. If read literally, the logical formula reads "there exists some thing x such that x is a knife and Brutus stabs Caesar with it". The existential quantifier is required due to the indefinite reading of "a knife" (Kearns, 2000). Here "a knife" signals that we have an object that is known to be a member of the set of knives, but no particular, uniquely identifiable knife is singled out[11]. The third argument place introduced to the function STAB is used to indicate the object that was used to perform the action and can be viewed as one of the many contextual coordinates that are required to

[11]This particular reading reflects a *non-specific* indefinite (Kearns, 2000) term. The issues surrounding definite and indefinite descriptions are discussed later in section 3.4.

fully describe the STAB action. Additional prepositional phrases would potentially require more argument places on each predicate.

The same comments for the first order predicate calculus representation apply to the intensional logic expression seen in (17). Existential quantification is introduced in order to create the indefinite description of “a knife”, and additional argument places are created for stab’ in order to accommodate the prepositional phrase. The extensional reading of both stab’ and knife’ make the entire expression equivalent to the first order predicate calculus representation in (16).

Turning to conceptual structures, in representation (18), the empty *Thing* constituent is filled by the *Thing* KNIFE. However, there are several ways that this English sentence may have been realized using conceptual structures. Assuming there was a primitive conceptual function STAB that relates two *Things*, we could obtain the structure as expressed in:

$$\begin{bmatrix} \text{STAB([BRUTUS], [CAESAR])} \\ \text{[WITH[KNIFE]]} \end{bmatrix} \tag{19}$$

This new structure contains a semantic function which is much closer to the one found in the first order predicate calculus representation. The primitive function STAB may also be defined to accept three arguments to give:

$$\begin{bmatrix} \text{STAB([BRUTUS], [CAESAR], [KNIFE])} \end{bmatrix} \tag{20}$$

While (20) contains the same meaning as (19), the definition of STAB in (19) provides the greatest flexibility, since additional circumstances beyond the basic action can simply be *fused* to the main *Event* constituent. Thus a complex sentence such as “Brutus stabbed Caesar with a knife at noon on the steps” becomes:

$$\begin{bmatrix} \text{STAB([BRUTUS], [CAESAR])} \\ \text{[WITH[KNIFE]]} \\ \text{[AT([NOON])]} \\ \text{[ON([STEPS])]} \end{bmatrix} \tag{21}$$

It is worthwhile to note at this point as well, that each major syntactic structure is mapped onto a conceptual constituent (Jackendoff, 1990). To see how this works, consider the following syntactic structure that relates to our original sentence “Brutus stabbed Caesar with a knife at noon on the steps”:

$$[_{NP} \text{ Brutus }]\ [_{VP} \text{ stabbed } [_{NP} \text{ Caesar}]\ [_{PP} \text{ with a knife}]\ [_{PP} \text{ at noon}]\ [_{PP} \text{ on the steps}]] \tag{22}$$

By comparing the syntactic structure in (22) to the conceptual structure in (21), we can observe the mappings in detail. The NPs Brutus and Caesar become *Thing* constituents which are directly incorporated into the conceptual function STAB. As the verb STAB is considered to be the *head* of the VP, it becomes the main function in the conceptual structure. Since the PPs serve to *modify* the circumstances of the main action, they become conceptual *Places* that are fused with the main function[12].

[12] For the purposes of this paper, manner, time and place elements are related to the action.

2.6 Anaphoric Expressions

Another complexity that exists in natural language is the notion of *anaphora*. In anaphoric expressions, *pronouns* or other referring expressions are used to refer to a previously mentioned object (McCoy and Strube, 1999). Consider the sentence "Juliet stabbed herself"[13]. Here the reflexive "herself" points back to the object "Juliet". Valid first order predicate calculus, intensional logic and conceptual structure representations become:

$$\text{STAB(juliet, juliet)} \tag{23}$$

$$\check{}\text{stab'(juliet', juliet')} \tag{24}$$

$$[\ \text{STAB}([\ \text{JULIET}\ ]^{\alpha},\ [\ \alpha\ ])\ ] \tag{25}$$

In the first order predicate calculus representation (23) the constant term "juliet" is used twice, filling both argument places for the predicate STAB. Unfortunately, this representation does not provide clues as to when a pronoun may be used in the realized syntactic construct. The literal meaning of this expression would be "Juliet stabbed Juliet", an English statement that is completely unnatural and potentially ambiguous in situations where there may be more than one Juliet[14].

As becomes obvious from the extensional reading of stab' in the intensional logic expression in (24), the representation is equivalent to the first order predicate calculus expression in (23). However, to solve some of the problems with knowing when to use a pronoun, an alternative expression may be employed:

$$\exists \text{x}[\ \text{x} = \text{juliet'} \wedge \check{}\text{stab'(juliet', x)}\ ] \tag{26}$$

Montague (1974b) allows for the identity operator = in intensional logic. This operator assigns the characteristic function of a predicate or designator to a variable. Hence, the semantic value of x takes on the denotation of juliet'. The advantage of (26) over (24) is that the x can give us clues where a pronoun may be used in a syntactic representation, albeit at the price of introducing an existentially quantified variable.

Turning to conceptual structures, the α in representation (25) is used to indicate binding between conceptual constituents (Jackendoff, 1990). In this instance, the superscripted α attached to the *Thing* JULIET signals that any further use of the symbol α will be taken to refer to the *Thing* JULIET. Such variable assignments provide a way of referring back to a previously introduced conceptual constituent without introducing ambiguity. For example, with the sentence "a Roman stabbed a Roman", there is ambiguity involved as to whether there is more than one Roman participating in the event. Without using variables to indicate references to previously introduced constituents, we would have the following structure:

$$[\ \text{STAB}([\ \text{ROMAN}\ ],\ [\ \text{ROMAN}\ ])\ ] \tag{27}$$

With (27), ambiguity arises as to whether the second instance of the *Thing* ROMAN is related to the first. The use of Greek symbols in representation (25) prevents this problem

[13]Sadly, the world could never know that Juliet became a Montague.

[14]To resolve this type of ambiguity in the first order predicate calculus representation, the reader is reminded that separate constant terms would be created for each Juliet, say "juliet1" and "juliet2".

from occurring, and provides us hints that further instances of Greek symbols may be appropriate for replacement with pronouns or reflexives[15].

2.7 Implication

Consider the instance where we wish to relate the fact that the action of Brutus stabbing Caesar resulted in his death. The English sentence “Brutus stabbed Caesar to death” conveys this information. The various representations become:

$$\text{STABTODEATH(brutus, caesar)} \tag{28}$$

$$\begin{array}{l}\exists e \exists d \exists x[\ \check{}\text{knife'}(x) \wedge e = \check{}\text{stab'}(\text{brutus', caesar'}, x) \wedge \\ d = \check{}\text{die'}(\text{caesar'}) \wedge \check{}\text{cause'}(e, d)\ ]\end{array} \tag{29}$$

$$\left[\ \text{CAUSE}\left(\begin{array}{l}[_{Event}\ \text{STAB}([\ \text{BRUTUS}\ ], [\ \text{CAESAR}\ ]^{\alpha})],\\ [_{State}\ \text{BE}([\ \alpha\ ], [\ \text{DEAD}\ ])]\end{array}\right)\ \right] \tag{30}$$

The STABTODEATH first order predicate calculus construction in (28) relates the fact that Brutus stabbed Caesar to death. Unfortunately, we are forced to create a predicate that fuses the notions of “stab” and “death” together into an inseparable entity. A more natural tendency would be to create an expression such as:

$$\text{STAB(brutus, caesar)} \wedge \text{DIE(caesar)} \tag{31}$$

While (31) follows our sensibilities about keeping the two predicates separate, the logical $\wedge$ does not properly express the causal connection between the two actions. In other words, the STAB predicate is interpreted completely separately from the DIE predicate, and in no way does the expression imply that STAB caused DIE to occur (Jurafsky and Martin, 2000).

The intensional logic representation in (29) allows us to express fine-grained detail over the exact causes of Caesar’s death. This is accomplished with the identity operator that allows a variable to represent a characteristic function (or any other complex type) (Dowty, 1981). The cost of this particular implementation is the introduction of three existentially qualified variables. However, we could have allowed for functions as arguments, thus reducing the expression to:

$$\exists x[\ \check{}\text{knife'}(x) \wedge \check{}\text{cause'}(\ \check{}\text{stab'}(\text{brutus', caesar'}, x), \check{}\text{die'}(\text{caesar'})\ ] \tag{32}$$

Additionally, had we wanted to express the fact that it was the knife that caused Caesar’s death, we could do so by manipulating the order of the arguments in (32) accordingly.

Conceptual structures provide for a great deal of control over subtle linguistic differences that may appear in English while maintaining a greater degree of clarity. The expression in (30) indicates that the *act* of stabbing Caesar is what resulted in Caesar’s death in the sentence “Brutus stabbed Caesar to death”. With this representation, the *Event* related to the stabbing of Caesar is clearly marked as the CAUSE of the death of Caesar. The representation of “Caesar died from the stab wound received from Brutus” would become:

[15]Strictly speaking, the use of Greek symbols should be employed with every constituent. The lack of a Greek symbol on a constituent signals a new and unique object.

$$\left[\text{CAUSE} \left(\begin{array}{l} [_{Event} \text{ CAUSE} \left(\begin{array}{l} [_{Event} \text{ STAB}([\text{ BRUTUS }],[\text{ CAESAR }]^{\alpha})], \\ [_{Event} \text{ WOUND}([\ \alpha\ ])] \end{array} \right), \\ [_{State} \text{ BE}([\ \alpha\ ], [\text{ DEAD }])] \end{array} \right) \right] \quad (33)$$

Again, with (33), the structured layout of the representation helps the reader to understand what is the ultimate CAUSE of the death of Caesar.

3 Comparing Expressiveness

Having explored the basic principles behind each formalism, we now turn to evaluating their expressiveness. As stated previously, expressiveness is measured in terms of linguistic coverage, precision and accuracy. In this section, a number of more complex linguistic phenomena will be explained and expressed using the first order predicate calculus, intensional logic and conceptual structures[16]. The overall goal of this section is to highlight the strengths and weaknesses of each formalism.

3.1 Tense

In the real world, facts about objects change depending on time. To understand an expression such as "Brutus stabs Caesar", we need to know when this statement applies (Dowty, 1981). With the treatment of English syntax, a variety of syntactic devices, such as auxiliary verbs and various verb suffixes, signal that interpretation of the current expression is to be evaluated relative to different points in time (Moss and Tiede, 2006; Saeed, 2003; Reichenbach, 1947). For this type of analysis to work, time needs to be viewed as a series of "instants in time" with an ordering on how each one follows from the other (Saeed, 2003). Using this analysis, we can classify an event as occurring before the current time (the past), at the current time (the present) or after the current time (the future)[17] (Reichenbach, 1947; Saeed, 2003). This particular analysis allows us to define what are called "simple tenses". Turning to concrete linguistic examples, the verb in "Brutus stabs Caesar" lacks mark-up that would suggest a different time for interpretation. This unmarked form is indicative of the *present tense*. Other tenses such as the *past* or *future* tense rely on overt mark up of the English syntax to indicate that a different temporal time frame should be used when interpreting the sentence. For example, the verb in "Brutus stabbed Caesar" has morphological markings which suggest interpretation should take place at an earlier temporal location.

Up to this point in the paper, we have treated all of our expressions as applying in some past temporal state in order to force a past tense interpretation of various expressions. We will change that notion now so that unless otherwise indicated, all statements reflect the current time or *present tense*.

3.1.1 Simple Past and Future Tenses

The following expressions attempt to capture the past tense sentence "Caesar died":

[16] In instances where an unmodified form of the first order predicate calculus is unable to express intended meaning, specially modified versions of the formalism will be used.

[17] Reichenbach (1947) refers to the current time as the *speech event* time. In other words, the current time corresponds to when the phrase is uttered, read, or otherwise produced

$$\exists e(\ \text{DIE(caesar, e)} \wedge \exists t(\ \text{TLOC(e, t)} \wedge t < s\ )\ ) \tag{34}$$

$$\text{P[die'(caesar')]} \tag{35}$$

$$\begin{bmatrix} \text{DIE([CAESAR])} \\ \text{[AT}_{Temp}\text{([PAST])]} \end{bmatrix} \tag{36}$$

The model in the first order predicate calculus representation in (34) has changed slightly in order for us to capture tenses. Presumably, the domain of entities A contained only individuals. We now expand it to include events and instances of time (Blackburn and Bos, 2003). With this new model in mind, we find that the DIE predicate has changed to relate the death of Caesar to an entity e, which is understood to be the *event* that the predicate refers to. The TLOC predicate equates the death event e to the existentially quantified variable t, which is understood to represent the *time* at which event e occurs. A second time, the *current time* is indicated with the s term. The past tense reading of the predicate is obtained through the notation $t < s$, which has the effect of saying that t (the event time) occurs at an earlier point in time than s (the current time). When taken together, the entire statement reads: Caesar's death event e occurred at a time t that was earlier than the current time s. Although this expression captures the "essence" of the past tense, it does so at the cost of creating an ontologically complex entity domain, since we now have to allow both events and times in the domain of individuals. However, this mechanism easily expresses future tense readings, such as "Caesar will die":

$$\exists e(\ \text{DIE(caesar, e)} \wedge \exists t(\ \text{TLOC(e, t)} \wedge t > s\ )\ ) \tag{37}$$

The only change that is evident in representation (37) is that the time t specified is *greater* than the current time s. This has the effect of saying that the event in question will take place in the future. This particular treatment of simple past and future tenses is satisfactory in terms of the precision and accuracy needed to capture the meaning of *simple* tense statements.

The intensional logic expression in (35) introduces the P operator. Recall that in the model for intensional logic, $<$ was a partial ordering on T (the set of all instances of time). In the intensional logic language, t is understood to be the current time while any other t' is understood to be at a point before or after t (Dowty, 1981). The P operator works using the partial ordering on T, by evaluating die'(caesar') at time coordinate t', where $t' < t$. In other words, if there is any point in time prior to the current time where die'(caesar') would evaluate to true, then the entire expression of P[die'(caesar')] will be true. This provides a simple past tense reading. An additional operator for the simple future tense also exists, making the sentence "Caesar will die" possible:

$$\text{F[die'(caesar')]} \tag{38}$$

The expression in (38) will be true if there exists any value of die'(caesar') that evaluates to true at any point in time t' such that $t' > t$. Obvious difficulties arise if our current time coordinate t happens to be at the start of time or at the end of the set of all times. For example, using the future tense operator F at the end of time will result in an undefined value, as will employing the past tense operator P at the beginning of time. However, Dowty

(1981) rationalizes this issue away by stating that the domain of times T can be viewed as an infinite set. Since reality says that no human being was alive at the beginning of time to utter a past tense phrase, and since it is unlikely that any human will exist to utter a future tense phrase at the end of time, the problem of using F or P at either end of the time continuum is, practically speaking, a non-issue. As with the first order predicate calculus representations, this particular treatment is both precise and accurate for use with simple past and perfect tenses.

Turning to conceptual structures, the expression in (36) is a likely candidate to express a past tense event. Jackendoff's discussions do not explicitly cover a treatment of tense, although there are hints as to how this might be accomplished. Jackendoff (1983) introduces the AT_{Temp} function, which indicates an exact temporal location that an event occurred. In this particular expression, we will use this function to indicate (rather abstractly) that the event in question occurred at some point in temporal space we know to have already passed. Other temporal locations may include FUTURE, TOMORROW or YESTERDAY, each of which can be interpreted to be in the past or future respectively. The expression of "Caesar will die" becomes:

$$\left[\begin{array}{l} \mathrm{DIE}([\ \mathrm{CAESAR}\ ]) \\ [\ \mathrm{AT}_{Temp}([\ \mathrm{FUTURE}\ ])\ ] \end{array}\right] \tag{39}$$

Conceptual structures allow for the same sort of precision and accuracy that both the first order predicate calculus and intensional logic expressions allowed.

3.1.2 Progressive and Perfect Tenses

More complex tenses may be obtained through various combinations of tense and *aspect*. In linguistic terms, grammatical aspect is concerned with how the action is distributed over a period of time (Saeed, 2003). In simple tenses, the start or end point of the action being described is either unimportant or unknown. For example, the sentence "Caesar went to the Senate" does not give any indication as to whether the action is complete or not, it just provides information that the action occurred sometime in the past. In this example, we say that the conjugation of the verb "go" has an indefinite aspect, and when combined with a past tense operator it gives rise to the *simple past* tense. In the previous section, these simple tenses can easily be expressed by stating the event time t relative to the current time s. However, other aspects exist that give rise to different readings that cannot be expressed using this simple analysis of tenses. Consider the sentence "Caesar was going to the Senate". In this instance, the verb "go" presents an aspect that the action is ongoing but may not yet be complete. The only information available is that the start of the action occurred sometime in the past. This particular aspect suggests an action that is *continuing*, and when combined with a past tense operator gives rise to the *past progressive* tense (Saeed, 2003). Different still is the verb "go" in the sentence "Caesar had gone to the Senate". With this example, the aspect suggests the action is *complete*, and combines with a past tense operator to give rise to the *past perfect* tense (Saeed, 2003).

Each of the aspects discussed above require that we extend the analysis of tenses to include a *reference* time r (Reichenbach, 1947). This point in time may be before or after the speech time and before or after the actual event time. In order to obtain progressive or perfect tenses, we need to know whether or not the event was complete or ongoing at the reference time in the past or future. This is accomplished by changing the associated time and ordering of t, r or s in a given expression. For example, consider the sentence

"Brutus had stabbed Caesar". With this example, the ordering of the event time t, the reference time r and the speech time t would be t < r < s. Essentially this tells us that the event t was complete before some reference point r, and that the event was in the past since r < s. Many different combinations of tenses and aspects are possible with this model (Reichenbach, 1947; Moss and Tiede, 2006).

To see how each formalism deals with different tenses and their aspects, consider the past perfect tensed statement "Caesar had gone to the Senate" represented below:

$$\exists \mathrm{e}(\ \mathrm{GO}(\mathrm{caesar},\ \mathrm{senate},\ \mathrm{e}) \wedge \exists \mathrm{t}(\ \mathrm{TLOC}(\mathrm{e},\ \mathrm{t}) \wedge \mathrm{t} < \mathrm{r} \wedge (\ \mathrm{r} < \mathrm{s}\ )\ )\ ) \tag{40}$$

$$\mathrm{PP}[\ \mathrm{go'}(\mathrm{caesar'},\ \mathrm{senate'})\ ] \tag{41}$$

$$\begin{bmatrix} \mathrm{GO}([\ \mathrm{CAESAR}\ ],\ [\ \mathrm{TO}([\ \mathrm{SENATE}\ ])\ ]) \\ [\ \mathrm{AT}_{Temp}([\ \mathrm{BEFORE}([\ \mathrm{PAST}\ ])\ ])\ ] \end{bmatrix} \tag{42}$$

The first order predicate calculus expression seen in (40) has introduced the symbol r to represent the *reference time*, while the symbols s and t still represent the current time and event time respectively. This particular expression directly incorporates the system described by Reichenbach (1947). Knowing this, representation (40) has changed slightly to incorporate reference time r. The GO predicate has been expanded in order to relate the variable e to the action of Caesar going to the Senate, while the TLOC predicate states that event e and time t are related. The rest of the statement focuses on temporally locating the event time t relative to a reference r and the current time s. As in the previous example, the ordering of t < r < s provides the perfect past reading.

The intensional logic representation in (41) introduces no new operators. Instead it recursively applies the P operator to create the term PP. As seen in the previous section, the P operator forces the evaluation of the expression to be performed at some earlier time coordinate. The repeated application of the P operator to make PP directs the evaluation to occur at some point in time before the point in time we are being directed to, thus providing the *past perfect* tense (Dowty, 1981). Additionally, by the same mechanism, future perfect tenses can be obtained by modifying the expression with the operator combination FP. However, while repeated applications of the F or P operator are allowed in infinite combinations (for example, FFF, PPP, FPF, etc), there is no way to use this particular treatment to represent any of the progressive tenses or a proper present perfect tense (Dowty, 1981).

With the conceptual structure representation in (42), a new function BEFORE is introduced to signal that the event occurred at some time before a reference point in the past. As in the simple tense representation seen in section 3.1.1, this particular function argument structure is not discussed directly by Jackendoff, but may be possible if we assume there is an ontological category called *Time*. Jackendoff (1983) makes mention that other ontological categories such as *Time* may be possible beyond the relatively well defined categories of *Thing*, *State*, *Event*, etc. However, he does not suggest any treatment beyond how one may handle *Events* and *States* within the spatial domain. If we accept *Time* as a category, then it is also conceivable to accept *Duration* as a category. With these two categories, it would then be possible to model past progressive statements such as "Caesar had been going to the Senate" with:

$$\begin{bmatrix} \text{GO([CAESAR], [TO([SENATE])])} \\ [_{Duration} \text{ INCOMPLETE]} \\ \text{[AT}_{Temp}\text{([BEFORE([PAST])])]} \end{bmatrix} \tag{43}$$

This particular treatment provides the precision and accuracy needed to express a wide variety of tenses.

3.2 Modality

In linguistics, the term *modality* is used in relation to expressions that state the *possibility* or *necessity* of a particular proposition (Kearns, 2000; von Fintel, 2005). For example, "Brutus must kill Caesar" is a modal expression conveying necessity while, "Brutus may kill Caesar" is a modal expression conveying possibility. Modalities are classified with respect to the set of *possible worlds* under which a proposition might necessarily or possibly be true[18](von Fintel, 2005). Several different classifications of modality exist including the alethic, dynamic, deontic, epistemic and teleological modalities (von Fintel, 2005). For example, the expression "Brutus may kill Caesar" has two different readings depending on the type of modality that is intended. The *epistemic* interpretation conveys the meaning that Brutus had both the ability and intent to kill Caesar and the known facts of their relationship support the conclusion that Brutus may have taken this action. The *deontic* interpretation conveys the meaning that a behavioral code of conduct allows Brutus to kill Caesar, should Brutus wish to do so. Additional classifications of modality exist, each one reflective of a different set of possible worlds (von Fintel, 2005). For the purposes of this paper, the basic operation of necessity and possibility will be explored within the epistemic and deontic modalities.

3.2.1 Epistemic Modality

The epistemic modality expresses the necessity or possibility of a statement in relation to the set of facts at hand (Kearns, 2000; Saeed, 2003; von Fintel, 2005). For example, consider the statement "Brutus may kill Caesar". There are two possible values of truth that the English statement "may" suggests. We can imagine that in one instance Brutus does indeed kill Caesar and the denotation of "Brutus kills Caesar" will be true. However, there is a second instance where Brutus does not kill Caesar, and the denotation is false. In order to deal with this seemingly dual nature that "may" suggests, the denotation of the expression is obtained by quantifying the set of possible worlds that the statement operates within (von Fintel, 2005). With this particular analysis, statements that express possibility need only be true in an existentially quantified possible world. In other words, there need be at least *one* possible world in which the expression is true (Saeed, 2003). With statements that convey necessity such as "Brutus must kill Caesar" the set of possible worlds is universally quantified. In other words, the statement must be true across *all* possible worlds.

Turning to the expression of epistemic statements in each formalism, consider the statement "Rome must be ruled by an Emperor". The formal representations of each are given below:

[18]Possible world semantics simply says that there exist an infinite number of realities in which the *state of affairs* differs from the one in the actual world we live in (von Fintel, 2005).

$$\exists w(\text{ ACTUAL-WORLD}(w) \land \forall v(\text{ R}(w, v) \rightarrow \exists x(\text{ EMPEROR}(v, x) \land \text{RULE}(v, x, rome)))) \tag{44}$$

$$\Box \exists x[\text{ emperor'}(x) \land \text{rule'}(x, \text{rome'})] \tag{45}$$

$$\left[\text{ NECESSARY}_{+Ep}(\text{ RULE}([\text{ EMPEROR }], [\text{ ROME }]))\ \right] \tag{46}$$

The first order predicate calculus representation in (44) has changed slightly once again. First, the domain of entities A has changed to include the set of *possible-worlds*. Second, two new specially designed predicates have been created: the ACTUAL-WORLD predicate to pick out the real world, and an accessibility relation R that allows us to specify over what set of worlds a particular action should occur in (Blackburn and Bos, 2003). Finally, each predicate in the language now requires an extra argument place to indicate what world the truth value is relative towards. The entire expression may be read as: there exists a world w known as the *actual world*, and that accessing all other worlds v, there is some individual x in world v such that x is an Emperor and x rules Rome. To express the possibility in "Rome may be ruled by an Emperor", the expression becomes:

$$\exists w(\text{ ACTUAL-WORLD}(w) \land \exists v(\text{ R}(w, v) \rightarrow \exists x(\text{ EMPEROR}(v, x) \land \text{RULE}(v, x, rome)))) \tag{47}$$

In (47), as expected, the only change comes in the form of the variable v being existentially qualified instead of universally qualified. This change has the effect of saying that there exists a world w known as the *actual world*, and that accessing some other world v, there is some individual x in every world v such that x is an Emperor and x rules Rome. As expressed before, the distinction between necessity and possibility lies with which set of possible worlds are chosen for the statement.

Turning to the intensional logic expression in (45), the symbol $\Box$ is introduced. The $\Box$ binds the right hand side of the expression and forces the predicates emperor' and rule' to be examined for truth at every possible world coordinate w'. The expression is true if and only if the predicates are true under every single world coordinate (Dowty, 1981). This expresses the same effect that was captured with the first order predicate calculus representation seen in (44), albeit with a much more elegant expression. To express the possibility in "Rome may be ruled by an Emperor", the expression would become:

$$\Diamond \exists x[\text{ emperor'}(x) \land \text{rule'}(x, \text{rome'})] \tag{48}$$

The expression in (48) contains the $\diamond$ symbol, used to express possibility. In this instance, the $\diamond$ binds the left hand side of the expression and forces predicate emperor' and rule' to be examined for truth across possible world coordinates w'. However, in contrast to $\Box$, the $\diamond$ operator requires that the predicate hold under only one (or more) possible world coordinates (Dowty, 1981). Again, this analysis is consistent with the treatment of epistemic modality.

As in the case of tense, Jackendoff did not offer a specific set of structures for modality. However, the conceptual structure in (46) represents a possible candidate for expressing a modal statement. The subscripted *+Ep* is a semantic feature that describes the type of modal that is present. In general, the function of a given semantic feature is to indicate whether or not a particular property is to be applied to a function-argument structure

(Jackendoff, 1990). In this case, the feature *+Ep* signals that this is to be understood in an epistemic modality. Another modal function POSSIBLY could also be defined in order to express terms such as "Rome may be ruled by an Emperor":

$$\left[\text{ POSSIBLY}_{+Ep}(\text{ RULE([EMPEROR], [ROME])) }\right] \tag{49}$$

Again, the same analysis applies to POSSIBLY in that the *+Ep* semantic feature applies to the function argument structure.

3.2.2 Deontic Modality

The deontic modality expresses necessity and possibility relative to a set of possible worlds that agree with a behavioral code of conduct or set of rules (Kearns, 2000; Saeed, 2003; von Fintel, 2005). For example, the expression "Brutus must kill Caesar" expresses the fact that Brutus must adhere to a code of conduct in which he is ordered to kill Caesar. By contrast, the expression "Brutus may kill Caesar" expresses the fact that Brutus has permission to kill Caesar and may do so if he wishes to. It is possible for some English sentences with modal meanings to become ambiguous with respect to the type of modal context being created. For example, it is difficult to determine if the expression "Brutus must kill Caesar" should be interpreted in an epistemic or deontic context (is this statement a factual conclusion based upon the best available evidence, or is it law?). As pointed out by von Fintel (2005), usually the background context provides enough information to clarify the modal meaning. At the semantic level, this ambiguity is non-existant since unique modal operators are usually employed provide the necessary distinction. For example, while the $\Box$ operator in intensional logic conveyed epistemic necessity, a new symbol would be needed to convey deontic necessity.

Consider the expression of the following deontic statement "Brutus must kill Caesar" in each formalism:

$$\exists\text{w(ACTUAL-WORLD(w)} \land \forall\text{v(RDE(w, v)} \rightarrow \text{KILL(v, brutus, caesar)))} \tag{50}$$

$$\text{O[kill'(brutus', caesar')]} \tag{51}$$

$$\left[\text{ NECESSARY}_{+De}(\text{ KILL([BRUTUS],[CAESAR])) }\right] \tag{52}$$

The first order predicate calculus representation in (50) looks very similar to the epistemic representation in (44). However, instead of using a normal accessibility relation R, a relation which spans only deontic worlds is introduced called RDE. The necessity of the statement is captured by the fact that the statement is quantified over the set of all possible deontic worlds. To express deontic possibility, the quantification simply changes from universal to existential. The sentence "Brutus may kill Caesar" is represented by:

$$\begin{aligned}&\exists\text{w(ACTUAL-WORLD(w)} \land \exists\text{v(RDE(w, v)} \rightarrow \exists\text{x(EMPEROR(v, x)}\\&\land \text{RULE(v, x, rome))))}\end{aligned} \tag{53}$$

With the intensional logic expression in (51), we see the introduction of an O operator. Although not specified directly by Montague himself, the O operator has been suggested by other researchers as a way of indicating an *obligatory* statement (Saeed, 2003). In effect, the

O binds the right hand side of the expression such that the kill' predicate is evaluated relative to the set of deontic worlds. Furthermore the semantics of the O operator states that the denotation of the expression must be true across *all possible* deontic worlds. Additionally, the use of O requires there to be some type of ordering on possible worlds so that one can distinguish between different world types (deontic, epistemic, alethic, bouletic, etc)[19]. Although this type of world ordering wasn't directly mentioned by Montague, it is quite possible and practical in theory. To convey the permissive possibility of "Brutus may kill Caesar", a second operator Q is introduced that requires the sentence hold in only *one* deontic world[20]:

$$\text{Q[kill'(brutus', caesar')]} \tag{54}$$

Again, like the O operator, the Q operator allows for an elegant solution to expressing modality. Moreover, the use of $\Box$, $\Diamond$, O and Q allow for the precision and accuracy needed to convey different forms of modality. This treatment of adding additional operators that pick out other subsets of possible worlds can be easily expanded to account for other modal contexts such as the alethic, bouletic and teleological modalities.

The conceptual structure in (52) uses the same NECESSITY function argument structure seen in the epistemic statement of (46). The main difference is due to the semantic feature of *+De*, which forces a deontic interpretation of the sentence. This same semantic feature can also be applied to the POSSIBLY function to give the permissive reading of "Brutus may kill Caesar" as seen below:

$$\left[\text{ POSSIBLY}_{+De}\text{(KILL([BRUTUS],[CAESAR])) } \right] \tag{55}$$

Using the NECESSITY and POSSIBLY function argument structures along with semantic features such as *+De* or *+Ep* allows for the precision and accuracy needed to express modal contexts. Additional semantic features may be added to express other modal contexts as needed.

3.3 Quantification

In linguistics, *quantifiers* are used to describe the properties or states of a set of individuals (Barwise and Cooper, 1981). Consider for example the sentence "every conspirator stabs Caesar". This statement asserts that all the individuals that are "conspirators" are involved in the stabbing of Caesar. Changing the quantifier affects the reading of how many conspirators may have been involved. For example, "a few conspirators stab Caesar" suggests that a smaller set of individuals were involved in the stabbing of Caesar. In general, a quantifier is typically built from a determiner (such as all, many, none) and a noun (Barwise and Cooper, 1981). For example, the determiner in the first example sentence is "every" which combines with the noun "conspirator" to create the quantifier "every conspirator"[21].

[19]The alethic modality is also known as the *logical* modality and is concerned with possibility and necessity across all possible worlds while bouletic modality reflects truth according to a speaker's desires (von Fintel, 2005).

[20]Saeed (2003) suggests using an operator P to denote *possibility*. However, since P has been reserved for past tense statements in intensional logic, Q will be used instead.

[21]We will follow the treatment of quantifiers as described by (Barwise and Cooper, 1981). The reader should be aware that other analyses are possible in which a determiner such as *some* is defined as the quantifier. A DET + NOUN construction would form a *quantified expression.*

3.3.1 Every and Some

Consider the sentence "every conspirator stabs Caesar", represented in each formalism below:

$$\forall x(\text{ CONSPIRATOR}(x) \rightarrow \text{STAB}(x, \text{caesar}) \text{ }) \tag{56}$$

$$\forall x[\text{ conspirator'}(x) \rightarrow \text{stab'}(x, \text{caesar'}) \text{ }] \tag{57}$$

$$\left[\text{ STAB}\left(\left[\begin{array}{l} {}_{\mathit{Type}} \text{ CONSPIRATOR} \\ [{}_{\mathit{Amount}} \text{ ALL }] \end{array} \right] \quad , \quad [\text{ CAESAR }] \right) \right] \tag{58}$$

The representation in (56) should be familiar to every reader who knows the first order predicate calculus. In fact, this particular representation mirrors the intensional logic expression in (57). We will use this similarity to describe both expressions at the same time. Recall that both the first order predicate calculus model and the intensional logic model employed a variable assignment g. The universal quantifier ∀ binds a variable x such that in order for the statement to be true, the expression must evaluate to true for every value assignment the function g makes to the variable x (Dowty, 1981). This means that for every possible value of x, the expression CONSPIRATOR(x) → STAB(x, caesar) must be true as well. Exploring the predicates and their logical connectives gives the following reading: if the individual denoted by the variable x is a conspirator, then x stabs Caesar.

These particular representations look quite different from their English counterparts. At the logical level, a quantifier and a conditional statement are needed to model the fact that "every conspirator stabs Caesar". This logical treatment is quite different than the determiner + noun phrase combination that defines a quantifier. In other words, there appears to be no way of determining which sub-expression is responsible for the noun phrase "every conspirator" (van der Does and van Eijck, 1996). Based on our English language definition of how to build a quantifier, it appears that the isolation of "every conspirator" requires only the combination of the determiner "every" with the noun "conspirator". However, doing so in the first order predicate calculus is impossible. A first attempt would be to simply use a universal quantifier along with a predicate as in:

$$\forall x(\text{ CONSPIRATOR}(x) \text{ }) \tag{59}$$

Unfortunately, the expression in (59) does not relate to "every conspirator", instead it states that everything in the world is a conspirator. A higher level of abstraction is needed in order to capture the essence of "every conspirator" alone. As we have seen before, intensional logic allows us to use the λ-operator to abstract over the set of functions we can combine with a conditional statement to give us a meaning of "every conspirator" in isolation:

$$\lambda P[\text{ }\forall x[\text{ }{}^{\vee}\text{conspirator'}(x) \rightarrow {}^{\vee}P(x, \text{caesar'}) \text{ }] \text{ }] \tag{60}$$

With the expression in (60), we are able to refer abstractly to the characteristic functions of the predicates that may combine with the other well formed portion of the statement (Dowty, 1981). In essence, the P acts as a way of referring to any characteristic function which takes two entities and returns a truth value. Since "every conspirator" is only defined in contexts where a material implication is present, leaving the consequent abstractly defined allows us to capture "every conspirator", something that cannot be done in the first order predicate calculus alone (Dowty, 1981).

Turning to the conceptual structure representation in (58), the familiar STAB function argument structure returns. The difference comes in the form of the first argument. Instead of referring to a specific CONSPIRATOR *Token*, this argument refers to the *Type* of things that are a CONSPIRATOR (Jackendoff, 1983). In this way, we can view this particular structure as referring abstractly to the set of CONSPIRATORS that exist. An additional *Amount* is specified that indicates ALL of the things that are considered to be CONSPIRATORS are involved in this argument place. This particular treatment of using *Amounts* and *Types* allows for the easy expression of various numbers of things.

Moving on to quantifiers involving "some", consider the sentence "some conspirator stabs Caesar", represented in each formalism below:

$$\exists \text{x(CONSPIRATOR(x)} \wedge \text{STAB(x, caesar))} \qquad (61)$$

$$\exists \text{x[conspirator'(x)} \wedge \text{stab'(x, caesar')]} \qquad (62)$$

$$\left[\text{STAB}\left(\begin{bmatrix} {}_{\textit{Type}}\ \text{CONSPIRATOR} \\ [{}_{\textit{Amount}}\ \text{ONE} \,] \end{bmatrix} , [\ \text{CAESAR}\ ] \right) \right] \qquad (63)$$

As we would expect to see in (61) and (62), the universal quantifier $\forall$ has been replaced by the existential quantifier $\exists$. This difference has the effect of saying that there only need to be *one* value assignment to x that satisfies the expression for it to be considered true. This is in direct contrast with the universal quantifier which stated that *every* value assignment to x needed to satisfy the expression in order for it to be considered true. Additionally, the existential quantifier can be analyzed in both the first order predicate calculus and intensional logic owing to the fact that "some conspirator" is not involved in an implication relationship with another predicate. For example, for "some conspirator" the following first order predicate calculus representation is sufficient:

$$\exists \text{x(CONSPIRATOR(x))} \qquad (64)$$

The conceptual structure expression in (63) holds no surprises either. Again, the *Type* distinction is made along with an *Amount* of SOME to create the proper quantified expression "some conspirator". As we shall see in section 3.3.3, this type of treatment is very effective for dealing with any quantified amount.

3.3.2 Scopal Ambiguity with Quantifiers

Ambiguity in English can arise from a variety of factors, including polysemy (when a single word has multiple related meanings), syntactic variants (e.g. garden-path sentences[22]) and variation in quantifier scope (Kearns, 2000; Saeed, 2003). An example of this last case can been seen in the sentence "every conspirator stabs some senator". With this sentence, it is unclear as to whether there is a single existentially quantified senator that the universally

[22]Garden-path sentences are syntactic constructions that are formulated without punctuation such that a reader will begin to create an incorrect mental representation of the syntactic structure as they read each individual word. Eventually, other syntactic constituents further along in the sentence will not properly fit into the mental structure developed by the reader. These "parse errors" force the reader to go back, examine, and revise the syntactic structure in order to determine the correct meaning of the sentence (Ferreira *et al.*, 2001). An example of a garden-path sentence is "the blood spilled on the steps dried".

quantified set of conspirators stab, or whether every conspirator is stabbing a different existentially quantified senator (Kearns, 2000). A semantic formalism must be capable of expressing both in an unambiguous way. Consider the first reading of the example sentence:

$$\exists y \forall x(\text{ CONSPIRATOR(x)} \wedge \text{SENATOR(y)} \rightarrow \text{STAB(x, y) }) \tag{65}$$

$$\exists y \forall x[\text{ conspirator'(x)} \wedge \text{senator'(y)} \rightarrow \text{stab'(x, y) }] \tag{66}$$

$$\left[\ \text{STAB}\left(\ \begin{bmatrix} _{Type}\ \text{CONSPIRATOR} \\ [_{Amount}\ \text{ALL}\ ] \end{bmatrix}\ ,\ \ [\text{ SENATOR }]\ \right)\ \right] \tag{67}$$

Once again, we may analyze the first order predicate calculus and intensional logic expressions concurrently. When read literally, expressions (65) and (66) state that there exists some individual y (who is a senator), that every individual x (who is a conspirator) stabs. The reading of a *single* senator is due to the fact that the existentially quantified variable y appears outside of the scope of the universally quantified variable x. If we move the existentially quantified y variable *inside* of the scope of the universal quantifier, the reading changes:

$$\forall x \exists y(\text{ CONSPIRATOR(x)} \wedge \text{SENATOR(y)} \rightarrow \text{STAB(x, y) }) \tag{68}$$

In this instance, for every x who is a conspirator, there is some y who is a senator such that x stabs y. As we can see, although the English is ambiguous as to which reading is intended, the semantic representations in the first order predicate calculus and intensional logic are capable of distinguishing between the two depending on the placement of the quantifiers.

For the conceptual structure expression in (67), the sense of a singular senator being stabbed is achieved by leaving the SENATOR in the second argument place of the STAB function a *Token* value rather than a type. Had we wished to make it such that every conspirator stabbed *some* senator, not necessarily the same one, the same treatment with *Type* distinction can be employed with SENATOR:

$$\left[\ \text{STAB}\left(\ \begin{bmatrix} _{Type}\ \text{CONSPIRATOR} \\ [_{Amount}\ \text{ALL}\ ] \end{bmatrix}\ ,\ \ [\ _{Type}\ \text{SENATOR}\ ]\ \right)\ \right] \tag{69}$$

In (69), the amount of SENATOR is left unspecified, but since we are referring to a set of things, rather than a specific *Token*, the reading we are left with is that every conspirator stabbed something that was a senator, not necessarily the same SENATOR.

3.3.3 Few, A Few, Most and Many

Determiners such as *few*, *a few*, *most* and *many* cannot be represented in a formalism that only contains the ∀ and ∃ quantifiers (Barwise and Cooper, 1981). Consider the first order predicate calculus representations for the statement "every conspirator stabs Caesar":

$$\forall x(\text{ CONSPIRATOR(x)} \rightarrow \text{STAB(x, caesar) }) \tag{70}$$

In (70), an NP predicate CONSPIRATOR and a VP predicate STAB must combine with a logical connective and logical quantifier to create the proposition expressed by the English sentences. In section 3.3.1 we observed the fact that there were no directly identifiable sub-expressions in the first order predicate calculus representation that directly related to the English fragment "every conspirator". In essence the quantified noun phrase becomes lost in the logical form (van der Does and van Eijck, 1996). Despite this disappearance, it is possible to identify a "general format" that occurs in sentences that incorporate quantifiers such as "every" or "all". The universal quantifier is used to pick out all the objects in the world, and the use of the logical connective $\rightarrow$ restricts the members involved in the proposition to the type specified in the antecedent, namely those that are conspirators (Kearns, 2000). In effect, when paired with the universal quantifier, the logical connective $\rightarrow$ is what provides the sorting mechanism of all the objects in the world into conspirators or non-conspirators.

Unfortunately, with more restricted quantifiers such as "most", the use of this "general format" is not possible since there is no combination of logical quantifiers and logical connectives which can provide a precise definition of what "most" means (Barwise and Cooper, 1981; Kearns, 2000). Natural language quantifiers such as "most" require that we can identify the set of conspirators in the world and compare its size to the set of things that are non-conspirators. The universal quantifier paired with a logical connective only gives us the power to sort the entire set of objects into two sets: conspirators and everything else. We need a mechanism capable of separating real world objects into a series of classes at the outset (conspirators stabbing Caesar and conspirators not stabbing Caesar), and then a separate mechanism to compare class sizes (Kearns, 2000). In other words, the selection of *most conspirators* must be performed as one complete, directly identifiable step that uses tools that are lacking in both the first order predicate calculus and intensional logic.

Solutions to this problem come in several forms. One is the use of a *restricted quantifier notation* which can be demonstrated with the English statement "most conspirators stab Caesar" (Kearns, 2000):

$$\text{[Most x: CONSPIRATOR(x)] STAB(x, CAESAR)} \tag{71}$$

With the representation in (71), the determiner *most* and its corresponding NP predicate CONSPIRATOR have become a single object. Here "Most x" has specified the type of quantification while CONSPIRATOR(x) has restricted the range of the quantifier. Since no complex proposition is being formed with STAB, no logical connective is needed to express the remainder of the sentence. Unfortunately, restricted quantifier notation is limited since it provides no way of obtaining a truth-theoretic value (Kearns, 2000). In this example, the formal definition of "Most x" is left unspecified. Other formalisms such as *generalized quantifiers theory* use set theoretic definitions to express quantities such as most (Kearns, 2000). For example, "most conspirators stab Caesar" would be realized as:

$$|\text{CONSPIRATOR(x)} \cap \text{STAB(x, CAESAR)}| > |\text{CONSPIRATOR(x) - STAB(x, CAESAR)}| \tag{72}$$

With representation (72), *most* is expressed by the fact that the cardinality of the set of things that are conspirators who stab Caesar is larger than the cardinality of the set of things that are conspirators but did not stab Caesar. Both restricted quantifiers and generalized quantifiers theory allow us to express some quantified statements that intensional logic and the first order predicate calculus cannot.

Turning to Jackendoff's conceptual structures, we see there is no problem with restricted quantifiers. Consider the conceptual structure expressing the statement "most conspirators stabbed Caesar" below:

$$\left[\text{ STAB}\left(\begin{bmatrix} {}_{Type} \text{ CONSPIRATOR} \\ [_{Amount} \text{ MOST]} \end{bmatrix} \;,\; [\text{ CAESAR }] \right) \right] \tag{73}$$

As we can see, the "most" determiner is stated as an amount in the same way that we used ALL and SOME to describe universal and existential quantification. Jackendoff (1990) states that most conceptual constituents such as *Thing*, *Event*, *State*, etc can be quantified. Although he does not state directly that the *Amount* constituent is the recommended method of quantification, it fits well with how the conceptual structure model works. It is conceivable that all individuals have a concept that relates to quantified amounts such as many, few, a few, little, etc depending on the type or specific token that is involved in the relationship (Jackendoff, 1990). This model allows for the precise and accurate representation of all types of quantifiers.

3.4 Definite Descriptions

Definite descriptions are used to pick out real world objects (Kearns, 2000). Examples of definite descriptions include names such as Caesar and Brutus and demonstratives such as "that knife". According to Russell's theory of definite descriptions, in order for a description to be considered definite, there must be some thing in the real world that exists, and that thing must be unique (Kearns, 2000). In other words, the object must be existentially quantified and the reference must be unambiguous. For example, consider the statement "the Emperor of Rome is dead". Here there is only one person who is the Emperor of Rome and there must be something that is known as the Emperor of Rome. Associated with definite descriptions is the notion of *familiarity effects*. This phenomenon simply states that most objects must be first introduced or be familiar to a listener before a specific definite description can be used (Kearns, 2000). For example, the use of the term "the conspirator" presupposes that some particular individual has already been named in the past as a conspirator, and that the listener is familiar with their real world referent.

We have already seen treatment of some definite descriptions throughout this paper. For example, Caesar and Brutus are names that are definite descriptions of real world individuals. However, we have not explicitly given treatment to definite determiners such as "the" when combined with a noun phrase. Consider the statement "the Emperor of Rome is dead" represented in each formalism below:

$$\exists \text{x(ROMANEMPEROR(x)} \land \forall \text{y(ROMANEMPEROR(y)} \rightarrow \text{y = x)} \land \text{DEAD(x))} \tag{74}$$

$$\exists \text{x[} \forall \text{y[romanemperor'(y)} \leftrightarrow \text{y = x]} \land \text{dead'(x)]} \tag{75}$$

$$\left[\text{ BE([DEAD], [}_{Token} \text{ ROMANEMPEROR]) } \right] \tag{76}$$

The first order predicate calculus representation in (74) introduces the identity operator =. When read literally, this particular expression simply states that some Roman Emperor x exists, and that for all objects y, there is only one y that is x, the Roman Emperor, and that the Roman Emperor is dead. With this expression, *the* is expressed as the combination of the existential quantifier with a uniqueness constraint (Kearns, 2000). This has the overall effect of providing a definite description of an object. However, there are problems with this expression. With (74), we assume that the set of all Emperors of Rome is exactly one. This may seem to be a logical consequence, but if were dealing with "the dog", it becomes highly unlikely that only one dog existed in the entire world. Although the first order predicate calculus representation captures the notion of a definite object precisely and accurately, it does so by placing large restrictions on how many objects of the type in question must be in the model.

With the intensional logic representation in (75), we see a very similar structure to the one seen in the first order predicate calculus representation. Like the first order predicate calculus representation, the set of Emperors of Rome must only have a cardinality of one at the current index. However, due to the intensionality of the model, we are allowed to have larger numbers of Emperors at future, past or differing world coordinates without affecting the truth value of the current coordinate. In order to better accommodate how a user may formulate a uniqueness constraint, potential expansions to the intensional logic model include a "previous discourse coordinate" or a place coordinate (Dowty, 1981). These coordinates would allow for a more precise definition of what objects should be included inside the definition of any given set. For example, even though the set of all dogs is quite a large set, restricting membership based on a place coordinate would allow for a much more accurate truth-conditional statement.

With conceptual structures, Jackendoff theorizes that the *Token* versus *Type* distinction of an object is what makes the difference when designating a definite description (Jackendoff, 1983). A *Token* of a *Thing*, *State*, *Event*, etc picks out a specific object in the world. This can be seen in expression (76). Here a specific ROMANEMPEROR is known to be dead. Which ROMANEMPEROR this particular representation refers to is determined by what mental information the speaker has associated with the ROMANEMPEROR *Thing* in question. In other words, the speaker will have a definite ROMANEMPEROR in mind.

3.5 Indefinite Descriptions

Indefinite descriptions are used when referring to any member of a particular set that is not uniquely identifiable in context (Kearns, 2000). We have seen examples of indefinite descriptions throughout this paper, for example "Brutus stabbed Caesar with a knife". Here the indefinite description of "a knife" signals that we are talking about a member of the set of knives, without picking out a specific member of the set. Determiners such as "a" and "an" typically combine with singular noun phrases to signal indefiniteness. Additionally, indefinite descriptions may be classified as specific or non-specific, depending on whether an individual exists in reality, or no such entity exists (Kearns, 2000). For example, a specific indefinite reading may be found in the statement "Caesar is searching for a conspirator". In this instance we can imagine that Caesar is looking for Brutus, for example, someone he knows to be a conspirator. A non-specific indefinite reading is also possible if one considers an instance where no conspirator exists in reality, but Caesar *believes* that one is present.

Consider the specific treatment of the phrase "Caesar is searching for a conspirator", indicated below:

$$\exists \text{x}(\ \text{CONSPIRATOR(x)} \wedge \text{SEEK(caesar, x)}\ ) \tag{77}$$

$$\exists \text{x}[\ \text{conspirator'(x)} \wedge \text{seek(caesar', x)}\ ] \tag{78}$$

$$\left[\ \text{SEEK}([\ \text{CAESAR}\ ],\ [_{Token}\ \text{CONSPIRATOR}\ ])\ \right] \tag{79}$$

The first order predicate calculus statement seen in (77) adequately captures the meaning behind the English sentence, as we have seen in other examples throughout this paper. However, treatment of the second reading of this sentence becomes more problematic. Consider the instance where no such conspirator exists in reality. Since there is no specific entity that we are referring to in the actual world, there is no way of capturing Caesar's actions in a concise manner. It would be possible to construct a specific predicate expressing Caesar's actions in an intransitive way such as below:

$$\text{SEEKCONSPIRATOR(caesar)} \tag{80}$$

However, such a construction suggests that a different SEEK predicate would be needed for every type of individual in the world, both real and imagined. Although this would adequately solve the problem of seeking a non-existent individual, it would do so at the expense of requiring a larger number of predicates. For example, for every single class of object available, a different SEEK relation would be required (e.g. SEEKFRIEND, SEEKFOE, etc). If we are willing to accept this expense, then first order predicate calculus is both precise and accurate enough to allow us to express indefinite descriptions.

The intensional logic example in (78) is very similar in structure to the first order predicate calculus expression. However, the intensional logic formalism is able to express the second reading of the statement with ease. Consider the instance where no conspirator exists in reality, yet Caesar is searching for one:

$$\text{seek'(caesar', } {}^\wedge\lambda Q \exists x[\ \text{conspirator'}(x) \wedge {}^\vee Q(x)\ ]\ ) \tag{81}$$

In this particular example, Caesar is seeking some existentially quantified thing x which has a property of containing conspirator-like properties. To understand how we arrive at this reading, recall that the intensional operator binds everything to the right. This means that both the existentially quantified variable x and the predicate conspirator'(x) are intensional. The overall effect is that conspirator'(x) is evaluated relative to any possible world or time. Although there may be no conspirator in *this* reality, there may be one in a different reality. In the instance where no conspirator exists in any reality, the abstract predicate Q is introduced. While Q is extensional in nature since the ˇ cancels the intensional operator, it is defined abstractly so that we may describe a conspirator as a set of properties, rather than having to point out an actual individual. Although this particular analysis is both accurate and precise, the overall user-friendliness becomes severely diminished when attempting to analyze such complex constructions.

With the conceptual structure in (79), the *Thing* CAESAR is searching for a specific *Thing Token* that is a conspirator. As seen previously, the *Token* distinction specifies that the *Thing* CONSPIRATOR actually exists in the real world. In order to express the second reading where a conspirator may not exist in reality, a *Type* distinction may be made instead:

$$\left[\ \text{SEEK}([\ \text{CAESAR}\ ],\ [_{Type}\ \text{CONSPIRATOR}\ ])\ \right] \tag{82}$$

In (82) the use of a *Type* states that Caesar is looking only for a *Thing* that has the properties of being a conspirator. In much the same way that intensional logic only described the properties of the "thing" that Caesar is looking for, so too does this conceptual structure.

Without referring to a specific instance of a CONSPIRATOR, the *Type* distinction simply states that CAESAR is searching for something that fits with the concept of being a conspirator (Jackendoff, 1983). This particular analysis is both accurate and precise enough to allow us to express both specific and non-specific indefinite descriptions.

4 Conclusions

This paper examined three systems of semantic representation: the first order predicate calculus, intensional logic and conceptual structures. Each of these formalisms features certain aspects which makes it attractive for different reasons. The first order predicate calculus remains a favorite system of representation for computational linguists due to the abundance of automated tools available for inferencing and theorem proving. Montague's intensional logic adds intensional elements such as possible world coordinates and time coordinates directly to the underlying model and presents an elegant way of capturing the semantics of a modal or tensed expression without negatively impacting the precision and accuracy of extensional statements. Finally, Jackendoff's conceptual structures offers a meaning theoretic approach that is appealing for psychologists and linguists.

Using a number of simple linguistic phenomena as examples, we observed the expressiveness of each formalism. In nearly every case, each formalism proved to be capable of capturing meaning with sufficient accuracy and precision. However, the set of linguistic phenomena explored in this paper is by no means exhaustive. There are many more complex phenomena that occur in natural language that Jackendoff and Montague have not explicitly discussed. For example, Montague's original research did not provide treatment for some linguistic phenomena such as non-declarative sentences, relative clauses and questions. Similarly, Jackendoff's original research provides no treatment of conceptual functions beyond those that deal with the spatial domain. This does not mean that these formalisms are of no further interest for computational linguists. Each system of semantic representation has been - and continues to be - actively expanded to account for these and other phenomena. In fact, this paper has presented some of the extrapolations and extensions that several researchers have proposed for each of these formalisms. For example, the first order predicate calculus has been expanded several times to account for modal and tensed statements, resulting in the creation of several first order logic variants. Similarly, researchers have proposed extensions to Montague's intensional logic to account for phenomena such as non-declarative sentences, adverbs, questions, presupposition and propositional attitudes (Dowty, 1981). With Jackendoff's conceptual structures, similar expansions continue to occur, with some researchers even having proposed a restatement of Jackendoff's research in terms of model-theoretic constructs (Zwarts and Verkuyl, 1994). Although no single formalism may currently be capable of expressing the entirety of meaning that is possible in natural language, continued research in linguistic and computational fields offers expansions and extrapolations that enhance the precision, accuracy and coverage across a wide variety of phenomena.

Finally, although we have observed that the expressiveness of each formalism is sufficient to capture a large number of linguistic phenomena, the increasing notational complexity of each formalism makes them unusable by an average human being. While this particular aspect of semantic representation may not be of importance in computational tasks where the underlying semantic expressions are not revealed to the end-user, it becomes vital in instances where humans are directly responsible for the manipulation, input and interpretation of semantic information. Further evaluation of these semantic formalisms with respect to human usability aspects is warranted.

References

Ammicht, E., Fosler-Lussier, E., and Potamianos, A. (2007). Information seeking spoken dialogue systems - part I: Semantics and pragmatics. *IEEE Transactions on Multimedia*, 9(3):532–549.

Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

Blackburn, P. and Bos, J. (2003). Computational semantics. *Theoria*, 18(46):27–45.

Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th international conference on World Wide Web*, pages 757–766.

Bos, J. (2005). Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics IWCS-6*, pages 42–53.

Davis, S. and Gillon, B.S. (2004). Introduction. In S. Davis and B.S. Gillon (Editors), *Semantics: A Reader*, pages 1–111. Oxford University Press.

Donald, M. (2006). *A Metalinguistic Framework for Specifying Generative Semantics*. Master's thesis, Queen's University.

Dowty, D.R. (1981). *Introduction to Montague Semantics*. Synthese Language Library. D. Reidel Publishing Company.

Ferreira, F., Christianson, K., and Hollingworth, A. (2001). Misinterpretation of garden-path sentences: Implications for models of sentence processing and renalysis. *Journal of Psycholinguistic Research*, 30(1):3–20.

Haegeman, L. (2005). *Introduction to Government and Binding Theory*. Blackwell Publishing.

Jackendoff, R. (1983). *Semantics and Cognition*. Current Studies in Linguistics Series. The MIT Press.

Jackendoff, R. (1990). *Semantic Structures*. Current Studies in Linguistics. The MIT Press.

Jurafsky, D. and Martin, J.H. (2000). *Speech and Language Processing*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, New Jersey.

Kearns, K. (2000). *Semantics*. Modern Linguistics. Palgrave Martin.

Korfhage, R.R. (1966). *Logic and Algorithms with Applications to the Computer and Information Sciences*. John Wiley & Sons Inc.

Koyama, M., Morta, K., Mizobuchi, S., and Aoe, J. (1998). An efficient retrieval algorithm for case structures using TRIE. *IEEE International Conference on Systems, Man and Cybernetics*, 5:4572–4577.

Lewis, D. (1972). General semantics. In D. Davidson and G. Harman (Editors), *Semantics of Natural Language*, pages 169–218. D. Reidel Publishing Company.

Lønning, J.T. and Oepen, S. (2006). Re-usable tools for precision machine translation. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 53–56.

McCoy, K.F. and Strube, M. (1999). Generating anaphoric expressions: Pronoun or definite description? In D. Cristea, N. Ide, and D. Marcu (Editors), *The Relation of Discourse/Dialogue Structure and Reference*, pages 63–71. Association for Computational Linguistics, New Brunswick, New Jersey.

Montague, R. (1974a). The proper treatment of quantification in ordinary english. In R. Thomason (Editor), *Formal Philosophy*, pages 247–270. Yale University Press.

Montague, R. (1974b). Universal grammar. In R. Thomason (Editor), *Formal Philosophy*, pages 222–246. Yale University Press.

Moss, L.S. and Tiede, H.J. (2006). Applications of modal logic in linguistics. In P. Blackburn, J.F. van Benthem, and F. Wolter (Editors), *Handbook of Modal Logic*. Elsevier Science.

Partee, B.H. (1972). Some transformational extensions of Montague grammar. In B.H. Partee (Editor), *Montague Grammar*, pages 51–76. Academic Press.

Poole, G. (2002). *Syntactic Theory*. Modern Linguistics. Palgrave Martin.

Radford, A. (1997). *Syntactic Theory and the Structure of English*. Cambridge University Press.

Reichenbach, H. (1947). *Elements of Symbolic Logic*, chapter 51, pages 289–298. The MacMillan Company, New York, NY, USA.

Rubin, J.E. (1990). *Mathematical Logic: Applications and Theory*. The Saunders Series. Saunders College Publishing.

Saeed, J.I. (2003). *Semantics - Second Edition*. Blackwell Publishing.

Traat, M. and Bos, J. (2004). Unificational combinatory categorial grammar: combining information structure and discourse representations. In *Proceedings of the 20th international conference on Computational Linguistics*, page 296.

van der Does, J. and van Eijck, J. (1996). Basic quantifier theory. In *Quantifiers, logic and language*, volume 54 of *CSLI Lecture Notes*, pages 1–45. CSLI Publications.

von Fintel, K. (2005). Modality and language. In D.M. Borchert (Editor), *Encyclopedia of Philosophy Second Edition*. MacMillan.

Weischedel, R. (2006). Natural-language understanding at BBN. *IEEE Annals of the History of Computing*, 28(1):46–55.

Wierzbicka, A. (1996). *Semantics Primes and Universals*. Oxford University Press.

Zwarts, J. and Verkuyl, H. (1994). An algebra of conceptual structure; an investigation into Jackendoff's conceptual semantics. *Linguistics and Philosophy*, 17(1):1–28.