An Evaluation Semantics for DATR Theories

Bill Keller
School of Cognitive and Computing Sciences
The University of Sussex, Brighton, UK
billk@cogs.susx.ac.uk

Abstract

This paper describes an operational semantics for DATR theories. The semantics is presented as a set of inference rules that axiomatises the evaluation relationship for DATR expressions. The inference rules provide a clear picture of the way in which DATR works, and should lead to a better understanding of the mathematical and computational properties of the language.

1 Introduction

DATR was originally introduced by Evans and Gazdar (1989a; 1989b) as a simple, non-monotonic language for representing lexical inheritance hierarchies. A DATR hierarchy is defined by means of path-value specifications. Inheritance of values permits appropriate generalizations to be captured and redundancy in the description of data to be avoided. A simple default mechanism provides for concise descriptions while allowing for particular exceptions to inherited information to be stated in a natural way.

Currently, DATR is the most widely-used lexical knowledge representation language in the natural language processing community. The formalism has been applied to a wide variety of problems, including inflectional and derivational morphology (Gazdar, 1992; Kilbury, 1992; Corbett and Fraser, 1993), lexical semantics (Kilgariff, 1993), morphonology (Cahill, 1993), prosody (Gibbon and Bleiching, 1991) and speech (Andry et al., 1992). In more recent work, DATR has been used to provide a concise, inheritance-based encoding of Lexicalized Tree Adjoining Grammar (Evans et al., 1995). There are around a dozen different implementations of DATR in existence and large-scale DATR lexicons have been designed for use in a number of natural language processing applications (Cahill and Evans, 1990; Andry et al., 1992; Cahill, 1994). A comprehensive, informal introduction to DATR and its application to the design of natural language lexicons can found in (Evans and Gazdar, 1996).

The original publications on DATR sought to provide the language with (1) a formal theory of inference (Evans and Gazdar, 1989a) and (2) a model-theoretic semantics (Evans and Gazdar, 1989b). Unfortunately, the definitions set out in these papers are not general enough to cover all of the constructs available in the DATR language. In particular, they fail to provide a full and correct treatment of DATR's notion of 'global inheritance', or the widely-used 'evaluable path' construct. A denotational semantics for DATR that covers all of the major constructs has been presented in (Keller, 1995). However, it still remains to provide a suitably general, formal theory of inference for DATR, and it is this objective that is addressed in the present paper.

2 Syntax of DATR Theories

Let NODE and ATOM be finite sets of symbols. Elements of NODE are called *nodes* and denoted by N. Elements of ATOM are called *atoms* and denoted by a. Elements of ATOM* are called *values* and denoted by α, β, γ. The set DESC of DATR *value descriptors* (or simply *descriptors*) is built up from the nodes and atoms as shown below. In the following, sequences of descriptors in DESC* are denoted ϕ, ψ.

1. $a \in$ DESC for any $a \in$ ATOM
2. For any $N \in$ NODE and $\phi \in$ DESC*:

$N : \langle\phi\rangle \in$ DESC
"$N : \langle\phi\rangle$" $\in$ DESC
"$\langle\phi\rangle$" $\in$ DESC
"N" $\in$ DESC

Elements of DESC are either *atomic descriptors* (1) or else *inheritance descriptors* (2). Inheritance descriptors are further distinguished as being *local* (unquoted) or *global* (quoted). For $\phi \in$ DESC* a sequence of descriptors, an expression $\langle\phi\rangle$ is called a *path descriptor*. When each element of ϕ is atomic, then $\langle\phi\rangle$ is called a *path*, and denoted P.

For N a node, P a path and ϕ a (possibly empty) sequence of value descriptors, an equation of the form $N : P == \phi$ is called a (*definitional*) *sentence*. Informally, $N : P == \phi$ specifies a property of the node N, namely that the value of the path P is given by the sequence of value descriptors ϕ. A DATR *theory* $\mathcal{T}$ is a finite set of definitional sentences subject to the following requirement of *functionality*:

$$\text{if } N : P == \phi \in \mathcal{T} \,\&\, N : P == \psi \in \mathcal{T} \text{ then } \phi = \psi$$

Functionality thus corresponds to a semantic requirement that node/path pairs are associated with (at most) one value.

3 Inference in DATR

The problem of constructing an explicit theory of inference for DATR was originally addressed in (Evans and Gazdar, 1989a). In this work, an attempt is made to set out a logic of DATR statements. Consider for example the following rule of inference, adapted from (Evans and Gazdar, 1989a).

$$\frac{N_1 : P_1 == N_2 : P_2, \quad N_2 : P_2 == \alpha}{N_1 : P_1 == \alpha}$$

The premises are definitional sentences which can be read: "the value of path P_1 at node N_1 is (inherited from) the value of path P_2 at N_2" and "the value of path P_2 at node N_2 is α", respectively. Given the premises, the rule licences the conclusion "the value of path P_1 at node N_1 is α". Thus, the rule captures a logical relationship between DATR sentences. For a given DATR theory $\mathcal{T}$, rules of this kind may be used to deduce additional sentences as theorems of $\mathcal{T}$.

In contrast, the system of inference described in this paper characterizes a relationship between DATR expressions (i.e. sequences of descriptors) and the values they may be used to compute. As an example, consider the following (simplified) rule of the operational semantics:

$$\frac{\begin{array}{c}\text{if } N_1 : P_1 == \phi \in \mathcal{T} \text{ then} \\ \phi \Longrightarrow \alpha\end{array}}{N_1 : P_1 \Longrightarrow \alpha}$$

The rule is applicable just in case the theory $\mathcal{T}$ contains a definitional sentence $N_1 : P_1 == \phi$. It states that if the sequence of value descriptors ϕ on the right of the sentence evaluates to ($\Longrightarrow$) the sequence of atoms α, then it may be concluded that the node/path pair $N_1 : P_1$ also evaluates to α. Rules of this kind may be used to provide an inductive definition of an evaluation relation between DATR expressions and their values.

Both approaches to inference in DATR aim to provide a system of deduction that makes it possible to determine formally, for a given DATR theory $\mathcal{T}$, what follows from the statements in $\mathcal{T}$. The primary interest lies in deducing statements about the values associated with particular node/path pairs defined within the theory. Unfortunately, the proof rules described in (Evans and Gazdar, 1989a) are not sufficiently general to support all of the required inferences, and it is not obvious that the approach can be extended appropriately to cover all of the available DATR constructs. A particular problem concerns DATR's notion of non-local or *global* inheritance. The value expressed by a global inheritance descriptor depends on more than just the properties of the nodes specified by the definitional sentences of a theory. In fact, it only makes sense to talk about the value of a global descriptor relative to a given context of evaluation, or *global context*. Because the proof rules of (Evans and Gazdar, 1989a) just talk about DATR sentences, which do not make explicit reference to context, it is not possible to give a satisfactory account of the global inheritance mechanism.

The evaluation semantics described in the following sections provides a perspicuous treatment of both local and global inheritance in DATR. The rules capture the essential details of the process of evaluating DATR expressions, and for this reason should prove of use to the language implementer.

4 Local Inheritance

As a point of departure, this section provides rules of inference for a restricted variant of DATR which lacks both global inheritance and the default mechanism. This variant will be referred to as DATR$_L$. The syntax of DATR$_L$ is as given in section 2, except of course that the three forms of global inheritance descriptor are omitted. An example of a simple DATR$_L$ theory is shown next.

$$\begin{array}{ll}
\textbf{Noun}: & \langle\textbf{cat}\rangle == \textbf{noun} \\
 & \langle\textbf{suff}\rangle == \textbf{s} \\
 & \\
\textbf{Dog}: & \langle\textbf{cat}\rangle == \textbf{Noun} \\
 & \langle\textbf{root}\rangle == \textbf{dog} \\
 & \langle\textbf{sing}\rangle == \langle\textbf{root}\rangle \\
 & \langle\textbf{plur}\rangle == \langle\textbf{root}\rangle\ \textbf{Noun} : \langle\textbf{suff}\rangle
\end{array}$$

In this and all subsequent examples, a number of standard abbreviatory devices are adopted. Sets of definitional sentences with the same node on the left-hand side are grouped together and the node left implicit in all but the first given sentence. Also, a definitional sentence such as **Dog** : ⟨**cat**⟩ == **Noun** : ⟨**cat**⟩, where the path on the right is identical to that on the left, is written more succinctly as **Dog** : ⟨**cat**⟩ == **Noun**. Similarly, nodes on the right of a sentence are suppressed when identical to the node on the left.

The DATR$_L$ theory defines the properties of two nodes, **Noun** and **Dog**. The definitional sentences specify values for node/path pairs, where the specification is either *direct* (a particular value is exhibited), or *indirect* (the value is obtained by local inheritance). For example, the value of the node/path pair **Noun** : ⟨**cat**⟩ is specified directly as **noun**. In contrast, the node/path pair **Dog** : ⟨**cat**⟩ obtains its value indirectly, by local

Values :

$$\frac{}{\alpha \Longrightarrow \alpha}\ \mathit{Val}$$

Definitions :

$$\text{if } N : \langle\alpha\rangle == \phi \in \mathcal{T} \text{ then} \quad \frac{\phi \Longrightarrow \beta}{N : \langle\alpha\rangle \Longrightarrow \beta}\ \mathit{Def}$$

Sequences :

$$\frac{\phi \Longrightarrow \alpha \quad \psi \Longrightarrow \beta}{\phi\psi \Longrightarrow \alpha\beta}\ \mathit{Seq}$$

Evaluable Paths :

$$\frac{\phi \Longrightarrow \alpha \quad N : \langle\alpha\rangle \Longrightarrow \beta}{N : \langle\phi\rangle \Longrightarrow \beta}\ \mathit{Sub}$$

Figure 1: Evaluation Semantics for DATR$_L$

inheritance from the value of **Noun** : ⟨**cat**⟩. Thus **Dog** : ⟨**cat**⟩ also has the value **noun**. The value of **Dog** : ⟨**plur**⟩ is specified indirectly by a sequence of descriptors **Dog** : ⟨**root**⟩ **Noun** : ⟨**suff**⟩. Intuitively, the required value is obtained by concatenating the values of the descriptors **Dog** : ⟨**root**⟩ and **Noun** : ⟨**suff**⟩, yielding **dog s**.

We wish to provide an inductive definition of an *evaluation relation* (denoted $\Longrightarrow$) between sequences of DATR descriptors in DESC* and sequences of atoms (i.e. values) in ATOM*. We write

$$\phi \Longrightarrow \alpha$$

to mean that *the sequences of descriptors* ϕ *evaluates to the sequence of atoms* α. With respect to the DATR$_L$ theory above we should expect that **Dog** : ⟨**cat**⟩ $\Longrightarrow$ **noun** and that **Dog** : ⟨**root**⟩ **Noun** : ⟨**suff**⟩ $\Longrightarrow$ **dog s**, amongst other things.

The formal definition of $\Longrightarrow$ for DATR$_L$ is provided by just four rules of inference, as shown in figure 1. The rule for *Values* states simply that a sequence of atoms evaluates to itself. Another way of thinking about this is that atom sequences are basic, and thus cannot be evaluated further. The rule for *Definitions* was briefly discussed in the previous section. It permits inferences to be made about the values associated with node/path pairs, provided that the theory $\mathcal{T}$ contains the appropriate definitional sentences. The third rule deals with the evaluation of sequences of descriptors, by breaking them up into shorter sequences. Given that the values of the sequences ϕ and ψ are known, then the value of $\phi\psi$ can be obtained simply by concatenation. Note that this rule introduces some non-determinism, since in general there is more than one way to break up a sequence of value descriptors. However, whichever way the sequence is broken up, the result (i.e. value obtained) should be the same. The following proof serves to illustrate the use of the rules *Val*, *Def* and *Seq*. It establishes formally that the node/path pair **Dog** : ⟨**plur**⟩ does indeed evaluate to **dog s** given the DATR$_L$ theory above.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{dog} \Longrightarrow \mathbf{dog}}\ \mathit{Val}}{\mathbf{Dog} : \langle\mathbf{root}\rangle \Longrightarrow \mathbf{dog}}\ \mathit{Def} \qquad \dfrac{\dfrac{}{\mathbf{s} \Longrightarrow \mathbf{s}}\ \mathit{Val}}{\mathbf{Noun} : \langle\mathbf{suff}\rangle \Longrightarrow \mathbf{s}}\ \mathit{Def}}{\mathbf{Dog} : \langle\mathbf{root}\rangle\ \mathbf{Noun} : \langle\mathbf{suff}\rangle \Longrightarrow \mathbf{dog\ s}}\ \mathit{Seq}}{\mathbf{Dog} : \langle\mathbf{plur}\rangle \Longrightarrow \mathbf{dog\ s}}\ \mathit{Def}$$

The final rule of figure 1 deals with DATR's evaluable path construct. Consider a value descriptor of the form **A** : ⟨**B** : ⟨⟩⟩. To determine the value of the descriptor it is first necessary to establish what path is specified by the path descriptor ⟨**B** : ⟨⟩⟩. This involves evaluating the descriptor **B** : ⟨⟩ and then 'plugging in' the resultant value α to obtain the path $\langle\alpha\rangle$. The required value is then obtained by evaluating **A** : $\langle\alpha\rangle$. The rule for *Evaluable Paths* provides a general statement of this process: if a sequence of value descriptors ϕ evaluates to α and **N** : $\langle\alpha\rangle$ evaluates to β, then **N** : $\langle\phi\rangle$ also evaluates to β.

5 Global Inheritance

DATR's local inheritance mechanism provides for a simple kind of data abstraction. Thus, in the DATR$_L$ theory above, information about the plural suffix is stated once and for all at the abstract **Noun** node. It is then available to any instance of **Noun** such as **Dog** via local inheritance. On the other hand, information about the formation of singular and plural forms of *dog* must still be located at the **Dog** node, even though the processes involved are entirely regular. To overcome this problem, DATR provides a second form of inheritance: *global inheritance*. This section provides an evaluation semantics for a default-free variant of DATR with both local and global inheritance (DATR$_G$). A simple DATR$_G$ theory is shown below.

Noun : ⟨**cat**⟩ == **noun**
⟨**suff**⟩ == **s**
⟨**sing**⟩ == "⟨**root**⟩"
⟨**plur**⟩ == "⟨**root**⟩" ⟨**suff**⟩

Dog : ⟨**cat**⟩ == **Noun**
⟨**root**⟩ == **dog**
⟨**sing**⟩ == **Noun**
⟨**plur**⟩ == **Noun**

The new theory is equivalent to that given previously in the sense that it associates exactly the same values with node/path pairs. However, in the DATR$_G$ theory global inheritance is used to capture the relevant generalizations about the singular and plural forms of nouns in English. Thus, the sentence **Noun** : ⟨**sing**⟩ == "⟨**root**⟩" states that the singular form of any noun is identical to its root (whatever that may be). The sentence **Noun** : ⟨**plur**⟩ == "⟨**root**⟩"⟨**suff**⟩ states that the plural is obtained by attaching the (plural) suffix to the root.

To understand the way in which global inheritance works, it is necessary to introduce DATR's notion of *global context*. Suppose that we wish to determine the value of **Dog** : ⟨**sing**⟩ in the example DATR$_G$ theory. Initially, the global context will be the pair (**Dog, sing**). From the theory, the value of **Dog** : ⟨**sing**⟩ is to be inherited (locally) from **Noun** : ⟨**sing**⟩, which in turn inherits its value (globally) from the quoted path "⟨**root**⟩". To evaluate the quoted path, the global context is examined to find the current global node (this is **Dog**) and the value of "⟨**root**⟩" is then obtained by evaluating **Dog** : ⟨**root**⟩, which yields **dog** as required.

More generally, the global context is used to fill in the missing node (path) when a quoted path (node) is encountered. In addition, as a side effect of evaluating a global inheritance descriptor the global context is updated. Thus, after encountering the quoted path "⟨**root**⟩" in the preceding example, the global context is changed from (**Dog, sing**) to (**Dog, root**). That is, the path component of the context is set to the new global path **root**.

Let $\mathcal{T}$ be a DATR$_G$ theory defined with respect to the set of nodes NODE and the set of atoms ATOM. The set CONT of *(global) contexts of* $\mathcal{T}$ is defined as the set of all pairs of the form (N, α), for $N \in$ NODE and $\alpha \in$ ATOM*. Contexts are denoted by C. The evaluation relation $\Longrightarrow$ is now taken to be a mapping from elements of CONT × DESC* to ATOM*. We write

$$C \vdash \phi \Longrightarrow \beta$$

to mean that ϕ evaluates to β *in the global context* C.

To axiomatise the new evaluation relation, the DATR$_L$ rules are modified to incorporate the global context parameter. For example, the rule for *Evaluable Paths* now becomes:

$$\frac{C \vdash \phi \Longrightarrow \alpha \quad C \vdash N : \langle\alpha\rangle \Longrightarrow \beta}{C \vdash N : \langle\phi\rangle \Longrightarrow \beta}\; Sub_1$$

Two similar rules are required for sentences containing quoted descriptors of the forms "**N** : ⟨ϕ⟩" and "⟨ϕ⟩". Note that the context C plays no special role here, but is simply carried unchanged from premises to conclusion. The rules for *Values*, *Definitions* and *Sequences* are modified in an entirely similar manner. Finally, to capture the way in which values are derived for quoted descriptors three entirely new rules are required, one for each of the quoted forms. These rules are shown in figure 2.

Quoted Node/Path :

$$\frac{(N,\alpha) \vdash N : \langle\alpha\rangle \Longrightarrow \beta}{C \vdash \text{“}N : \langle\alpha\rangle\text{”} \Longrightarrow \beta}\; Quo_1$$

Quoted Path :

$$\frac{(N,\alpha) \vdash N : \langle\alpha\rangle \Longrightarrow \beta}{(N,\alpha') \vdash \text{“}\langle\alpha\rangle\text{”} \Longrightarrow \beta}\; Quo_2$$

Quoted Node :

$$\frac{(N,\alpha) \vdash N : \langle\alpha\rangle \Longrightarrow \beta}{(N',\alpha) \vdash \text{“}N\text{”} \Longrightarrow \beta}\; Quo_3$$

Figure 2: Evaluation of Quoted Descriptors

Consider for example the *Quoted Path* rule. The premise states that N : $\langle\alpha\rangle$ evaluates to β in the global context (N, α). Given this, the rule licences the conclusion that the quoted descriptor "$\langle\alpha\rangle$" also evaluates to β in any context with the same node component N. In other words, to evaluate a quoted path "$\langle\alpha\rangle$" in a context (N, α'), just evaluate the local descriptor N : $\langle\alpha\rangle$ in the updated global context (N, α). The rules dealing with global node/path pairs, and global nodes work in a similar way.

The following proof illustrates the use of the *Quoted Path* rule (Quo_2). It demonstrates that **Dog** : ⟨**sing**⟩ evaluates to **dog**, given the DATR$_G$ theory, and when the initial global context is taken to be (**Dog, sing**).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\mathbf{Dog},\mathbf{root}) \vdash \mathbf{dog} \Longrightarrow \mathbf{dog}}\,Val}{(\mathbf{Dog},\mathbf{root}) \vdash \mathbf{Dog} : \langle\mathbf{root}\rangle \Longrightarrow \mathbf{dog}}\,Def}{(\mathbf{Dog},\mathbf{sing}) \vdash \text{“}\langle\mathbf{root}\rangle\text{”} \Longrightarrow \mathbf{dog}}\,Quo_2}{(\mathbf{Dog},\mathbf{sing}) \vdash \mathbf{Noun} : \langle\mathbf{sing}\rangle \Longrightarrow \mathbf{dog}}\,Def}{(\mathbf{Dog},\mathbf{sing}) \vdash \mathbf{Dog} : \langle\mathbf{sing}\rangle \Longrightarrow \mathbf{dog}}\,Def$$

6 Path Extensions and Defaults

In DATR, values may be associated with particular node/path pairs either *explicitly*, in terms of local or global inheritance, or *implicitly* 'by default'. The basic idea underlying DATR's default mechanism is as follows: any definitional sentence is applicable not only to the path specified on its left-hand side, but also for any rightward extension of that path for which no more specific definitional sentence exists. Making use of defaults, the DATR$_G$ theory given above can be expressed more succinctly as shown next.

```
Noun :  ⟨cat⟩ == noun
        ⟨sing⟩ == "⟨root⟩"
        ⟨plur⟩ == "⟨root⟩" ⟨suff⟩
        ⟨suff⟩ == s

Dog :   ⟨⟩ == Noun
        ⟨root⟩ == dog
```

Here, the relationship between the nodes **Dog** and **Noun** has effectively been collapsed into just a single statement **Dog** : ⟨⟩ == **Noun**. This is possible because the sentence now corresponds to a whole class of implicit definitional sentences, each of which is obtained by extending the paths found on the left- and right-hand sides in the same way. Accordingly, the value of **Dog** : ⟨**cat**⟩ is specified implicitly as the value of **Noun** : ⟨**cat**⟩, and similarly for **Dog** : ⟨**sing**⟩ and **Dog** : ⟨**suff**⟩. In contrast, the specification **Dog** : ⟨**root**⟩ == **Noun** :

Values :

$$\frac{}{C \vdash \alpha \Longrightarrow_{\gamma} \alpha}\ \mathit{Val}$$

Definitions : if α is the longest prefix of $\alpha\gamma$ s.t. $N : \langle\alpha\rangle == \phi \in \mathcal{T}$, then

$$\frac{C \vdash \phi \Longrightarrow_{\gamma} \beta}{C \vdash N : \langle\alpha\gamma\rangle \Longrightarrow \beta}\ \mathit{Def}$$

Sequences :

$$\frac{C \vdash \phi \Longrightarrow_{\gamma} \alpha \quad C \vdash \psi \Longrightarrow_{\gamma} \beta}{C \vdash \phi\psi \Longrightarrow_{\gamma} \alpha\beta}\ \mathit{Seq}$$

Evaluable Paths :

$$\frac{C \vdash \phi \Longrightarrow_{\gamma} \alpha \quad C \vdash N : \langle\alpha\rangle \Longrightarrow_{\gamma} \beta}{C \vdash N : \langle\phi\rangle \Longrightarrow_{\gamma} \beta}\ \mathit{Sub}_1$$

$$\frac{C \vdash \phi \Longrightarrow_{\gamma} \alpha \ C \vdash \text{``}N : \langle\alpha\rangle\text{''} \Longrightarrow_{\gamma} \beta}{C \vdash \text{``}N : \langle\phi\rangle\text{''} \Longrightarrow_{\gamma} \beta}\ \mathit{Sub}_2$$

$$\frac{C \vdash \phi \Longrightarrow_{\gamma} \alpha \ C \vdash \text{``}\langle\alpha\rangle\text{''} \Longrightarrow_{\gamma} \beta}{C \vdash \text{``}\langle\phi\rangle\text{''} \Longrightarrow_{\gamma} \beta}\ \mathit{Sub}_3$$

Quoted Descriptors :

$$\frac{(N, \alpha\gamma) \vdash N : \langle\alpha\rangle \Longrightarrow_{\gamma} \beta}{C \vdash \text{``}N : \langle\alpha\rangle\text{''} \Longrightarrow_{\gamma} \beta}\ \mathit{Quo}_1$$

$$\frac{(N, \alpha\gamma) \vdash N : \langle\alpha\rangle \Longrightarrow_{\gamma} \beta}{(N, \alpha') \vdash \text{``}\langle\alpha\rangle\text{''} \Longrightarrow_{\gamma} \beta}\ \mathit{Quo}_2$$

$$\frac{(N, \alpha\gamma) \vdash N : \langle\alpha\rangle \Longrightarrow_{\gamma} \beta}{(N', \alpha) \vdash \text{``}N\text{''} \Longrightarrow_{\gamma} \beta}\ \mathit{Quo}_3$$

Path Extensions :

$$\frac{C \vdash N : \langle\alpha\gamma\rangle \Longrightarrow \beta}{C \vdash N : \langle\alpha\rangle \Longrightarrow_{\gamma} \beta}\ \mathit{Ext}$$

Figure 3: The Evaluation Semantics for DATR

⟨root⟩ does *not* follow 'by default' from the definition of **Dog**, even though it can be obtained by extending left and right paths in the required manner. The reason is that the theory already contains an explicit statement about the value of **Dog** : **⟨root⟩**.

The evaluation relation is now defined as a mapping from elements of CONT × DESC* × ATOM* (i.e. context/descriptor sequence/path extension triples) to ATOM*. We write:

$$C \vdash \phi \Longrightarrow_{\gamma} \alpha$$

to mean that ϕ evaluates to α in context C *given path extension* γ. When $\gamma = \epsilon$ is the empty path extension, we will continue to write $C \vdash \phi \Longrightarrow \alpha$.

A complete set of inference rules for DATR is shown in figure 3. The rules for *Values*, *Sequences* and *Evaluable Paths* require only slight modification as the path extension is simply passed through from premises to consequent. The rules for *Quoted Descriptors* are also much as before. Here however, the path extension γ appears as part of the global context in the premise of each rule. This means that when a global descriptor is encountered, any path extension present is treated 'globally' rather than 'locally'. The main change in the *Definitions* rule lies in the conditions under which it is applicable. The amended rule just captures the 'most specific sentence wins' default mechanism. Finally, the new rule for *Path Extensions* serves as a way of making any path extension explicit. For example, if **Dog** : **⟨cat⟩** evaluates to **noun**, then **Dog** : ⟨⟩ also evaluates to **noun** given the (explicit) path extension **cat**.

An example proof showing that **Dog** : **⟨plur⟩** evaluates to **dog s** given the DATR theory presented above is shown in figure 4.

7 Conclusions

The evaluation semantics presented in this paper constitutes the first fully worked out, formal system of inference for DATR theories. This fulfills one of the original objectives of the DATR programme, as set out in (Evans and Gazdar, 1989a; Evans and Gazdar, 1989b), to provide the language with an explicit theory of inference. The inference rules provides a clear picture of the way in which the different constructs of the language work, and should serve as a foundation for future investigations of the mathematical and computational properties of DATR. Although the rules abstract away from particular implementational details such as order of evaluation, they can be readily understood in computational terms and may prove useful as a guide to the construction of practical DATR interpreters.

Acknowledgements

The author wishes to thank Roger Evans, Gerald Gazdar and David Weir for suggestions and comments relating to this work.

References

Francois Andry, Norman Fraser, Scott McGlashan, Simon Thornton, and Nick Youd. 1992. Making DATR work for speech: lexicon compilation in SUNDIAL. *Computational Linguistics*, 18(3):245–267.

Lynne Cahill and Roger Evans. 1990. An application of DATR: the TIC lexicon. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pages 120–125.

Lynne Cahill. 1993. Morphonology in the lexicon. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 87–96.

Lynne Cahill. 1994. An inheritance-based lexicon for message understanding systems. In *Proceedings of the 4th ACL Conference on Applied Natural Language Processing*, pages 211–212.

Greville Corbett and Norman Fraser. 1993. Network morphology: a DATR account of Russian nominal inflection. *Journal of Linguistics*, 29:113–142.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\mathbf{Dog},\mathbf{root}) \vdash \mathbf{dog} \Longrightarrow \mathbf{dog}}\,\mathit{Val}}{(\mathbf{Dog},\mathbf{root}) \vdash \mathbf{Dog} : \langle\mathbf{root}\rangle \Longrightarrow \mathbf{dog}}\,\mathit{Def}}{(\mathbf{Dog},\mathbf{plur}) \vdash \text{``}\langle\mathbf{root}\rangle\text{''} \Longrightarrow \mathbf{dog}}\,\mathit{Quo}_2 \quad \dfrac{\dfrac{}{(\mathbf{Dog},\mathbf{plur}) \vdash \mathbf{s} \Longrightarrow \mathbf{s}}\,\mathit{Val}}{(\mathbf{Dog},\mathbf{plur}) \vdash \mathbf{Noun} : \langle\mathbf{suff}\rangle \Longrightarrow \mathbf{s}}\,\mathit{Def}}{(\mathbf{Dog},\mathbf{plur}) \vdash \text{``}\langle\mathbf{root}\rangle\text{''}\ \mathbf{Noun} : \langle\mathbf{suff}\rangle \Longrightarrow \mathbf{dog\ s}}\,\mathit{Seq}}{(\mathbf{Dog},\mathbf{plur}) \vdash \mathbf{Noun} : \langle\mathbf{plur}\rangle \Longrightarrow \mathbf{dog\ s}}\,\mathit{Def}}{(\mathbf{Dog},\mathbf{plur}) \vdash \mathbf{Noun} : \langle\rangle \Longrightarrow_{\mathbf{plur}} \mathbf{dog\ s}}\,\mathit{Ext}}{(\mathbf{Dog},\mathbf{plur}) \vdash \mathbf{Dog} : \langle\mathbf{plur}\rangle \Longrightarrow \mathbf{dog\ s}}\,\mathit{Def}$$

Figure 4: Proof utilising defaults

Roger Evans and Gerald Gazdar. 1989a. Inference in DATR. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, pages 66–71.

Roger Evans and Gerald Gazdar. 1989b. The semantics of DATR. In *Proceedings of AISB-89*, pages 79–87.

Roger Evans and Gerald Gazdar. 1996. DATR: A language for lexical knowledge representation. *Computational Linguistics* 22(2). Also available as CSRP 382, School of Cognitive and Computing Science, The University of Sussex and Technical report 95-15, ITRI, University of Brighton.

Roger Evans, Gerald Gazdar, and David Weir. 1995. Encoding lexicalized tree adjoining grammars with a nonmonotonic inheritance hierarchy. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pages 77–84.*

Gerald Gazdar. 1992. Paradigm function morphology in DATR. In Lynne Cahill and Richard Coates, editors, *Sussex Papers in General and Computational Linguistics*, number CSRP 239 in Cognitive Science Research Papers, pages 45–53. University of Sussex, Brighton.

Dafydd Gibbon and Doris Bleiching. 1991. An ILEX model for German compound stress in DATR. In *Proceedings of the FORWISS-ASL Workshop on Prosody in Man-Machine Communication.*

Bill Keller. 1995. DATR theories and DATR models. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 55–62.

James Kilbury. 1992. Pardigm-based derivational morphology. In Guenther Goerz, editor, *Proceedings of KONVENS 92*, pages 159–168. Springer, Berlin.

Adam Kilgariff. 1993. Inheriting verb alternations. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 213–221.