

# Human and Unhuman Commonsense Reasoning

Michael J. Maher

NICTA and University of NSW  
Sydney, Australia  
Michael.Maher@nicta.com.au

**Abstract.** Ford has introduced a non-monotonic logic, System **LS**, inspired by an empirical study of human non-monotonic reasoning. We define here a defeasible logic **FDL** based on Ford’s logic, and in doing so identify some similarities and differences between Ford’s logic and existing defeasible logics. Several technical results about **FDL** are established, including its inference strength in relation to other defeasible logics.

## 1 Introduction

From its earliest years, logic has been used both to model human reasoning and to express an idealization of human reasoning<sup>1</sup>. This work has focused largely on reasoning based on definitive, or certain statements. For example, in the classic syllogism there is no doubt that Socrates is a man, nor that, without exception, all men are mortal.

However, much of human language and reasoning involves statements that are not certain but rather are generalities that admit exceptions. The prototypical example is: “Birds fly”. An effort to reason formally and sensibly with such statements, inspired by the human ability to reason with “common sense”, has grown within Artificial Intelligence in the last 30 years. Human commonsense reasoning has been modelled and idealized in various ways, leading to the development and study of non-monotonic and defeasible reasoning.

There has been a proliferation of formalisms and a variety of semantics for them. Even a relatively simple formalism, non-monotonic inheritance networks (also known as inheritance networks with exceptions), has multiple interpretations based on clashing intuitions of researchers

---

<sup>1</sup> Interestingly, classical logic fails in some respects to model human reasoning or, to put it differently, humans often fail to reason in this ideal way [9, 20], and attempts continue to model human language and reasoning in logics.

[28]. However, relatively little empirical study of human reasoning on such problems has been carried out.

Recent studies by Ford and Billington [13, 11, 12] of human reasoning on problems that can be formulated as inheritance with exceptions have identified reasoning principles that appear to underlie rational human reasoning about such problems. Out of these studies, Ford formulated a logic System **LS** [10] reflecting the idealized human reasoning that these principles represent.

This paper addresses the preliminary design of a defeasible logic **FDL** based on Ford's logic and the underlying principles she identified. The logic fits into the framework for defeasible logics described in [4].

In the next section, two defeasible reasoning formalisms are outlined: non-monotonic inheritance networks and defeasible logic. Then, in Section 3, Ford's logic **LS** and some intuitions that lie behind it are presented. Section 4 defines the defeasible logic **FDL** based on these ideas and Section 5 presents some properties of this logic.

## 2 Defeasible Reasoning

Defeasible reasoning concerns reasoning where a chain of reasoning can be defeated (that is, not considered the basis of an inference) by another chain of reasoning (or, perhaps, several chains of reasoning). Thus defeasible reasoning is very similar to argumentation. In this section we discuss two formalisms for defeasible reasoning: defeasible logic and non-monotonic inheritance networks.

### 2.1 Defeasible Logic

The defeasible logics considered here are those considered in [1, 4, 7], developed from the original formulation of Nute [26]. There are various analyses of the semantics of these logics [16, 17, 22, 25] but here we will stick to a straightforward proof-theoretic view.

Defeasible logics have a comparatively simple syntax. The basic building blocks are literals (negated or unnegated atomic formulae) which may be arranged into rules. Rules consist of a single literal in the head, an arrow, and a collection of literals forming the body. Thus negation and arrows are the only syntactic operators, although we will find it convenient at times to view the bodies of rules to be constructed by a binary

conjunction operator. We use  $\neg$  as the negation symbol and  $\sim$  for the complementary operation which maps an atom to its negation and vice versa.

There are three different kinds of arrows, distinguishing three different kinds of rules. The arrow  $\rightarrow$  is used to denote definitive or *strict* rules: rules and inferences that are absolutely, certainly, always valid. For example, we might write

$$emu(X) \rightarrow bird(X)$$

denoting that we consider that every emu, without exception, is a bird.

The arrow  $\Rightarrow$  denotes a *defeasible rule*: rules that can be taken to be valid a lot of the time, but for which there are exceptions. They correspond to statements in English that are modified by words such as “generally”, “usually”, “typically”, “normally”, “mostly”, etc. For example,

$$bird(X) \Rightarrow flier(X)$$

denotes that birds are usually able to fly, while recognising that some birds, such as those covered in oil, might not fly.

The distinction between these two kinds of rules is common to several non-monotonic formalisms. The third rule, however is more unusual. It represents a reluctance or, stronger, an inability to allow some inferences. It cannot be used to make an inference based on its own form, and only describes situations where conflicting inferences are not allowed. The arrow is  $\rightsquigarrow$  and the rule is called a *defeater*. For example,

$$heavy(X) \rightsquigarrow \neg flier(X)$$

denotes that while we might consider that birds, in general, can fly (according to the previous rule) we do not allow this inference when the bird is heavy. On the other hand, we cannot use this rule for inference so we cannot conclude that something cannot fly just because it is heavy.

Defeasible logic also includes a set of *facts*, that is, literals that specified to be true. For example, we might have facts  $emu(tweety)$  and  $heavy(tweety)$ .

In addition, to resolve competing rules there is an acyclic binary relation among rules called the *superiority relation* and denoted by  $>$ .  $r_2 > r_1$  expresses that whenever both rules apply but have conflicting conclusions  $r_2$  should have priority over – or over-rule –  $r_1$ ; we should not

apply the inference associated with  $r_1$  and apply that associated with  $r_2$  (provided  $r_2$  is not a defeater). For example,

$$\begin{array}{ll}
r_1 : \text{bird}(X) & \Rightarrow \text{flier}(X) \\
r_2 : \text{nest\_on\_ground}(X), \text{animal}(X) & \Rightarrow \neg \text{flier}(X) \\
r_3 : \text{bird}(X) & \rightarrow \text{animal}(X)
\end{array}$$

$r_2 > r_1$

describes a situation where usually birds fly, and usually animals that nest on the ground do not fly, and when we find a bird that nests on the ground we should conclude that it does not fly, since  $r_2$  over-rides  $r_1$ .

Thus a *defeasible theory* is a triple  $(F, R, >)$  where  $F$  is a set of facts,  $R$  is a set of rules, and  $>$  is a superiority relation over  $R$ . Although rules are presented in a first-order style, we assume that they are grounded so that the proof theory operates over an essentially propositional language. A formal description of inference in a defeasible logic is, in general, more complicated than the above informal introduction suggests. We consider one deeply-studied defeasible logic **DL** [1] as an example. A proof is a sequence of tagged literals, where tags describe the strength of proof under consideration and whether the literal has been proved, or it has been established that the literal cannot be proved. For example,  $+\Delta p$  denotes that  $p$  can be proved using only the strict rules and  $-\Delta q$  denotes that all possibilities for proving  $q$  using only strict rules have been exhausted without proving  $q$ . Two other tags  $+\partial$  and  $-\partial$  refer to provability with respect to all rules.

The inference rules for  $\Delta$ , formulated as conditions on proofs, are given in Figure 1.  $R_s$  refers to the strict rules of  $R$ , and  $R_s[q]$  is the strict rules with head  $q$ . We will later use  $R_{sd}$  for the strict and defeasible rules of  $R$ , and  $R_{dd}$  for the defeasible rules and defeaters.  $A(r)$  refers to the

- $+\Delta$ ) If  $P(i+1) = +\Delta q$  then either
  - .1)  $q \in F$ ; or
  - .2)  $\exists r \in R_s[q] \forall a \in A(r), +\Delta a \in P[1..i]$ .
- $-\Delta$ ) If  $P(i+1) = -\Delta q$  then
  - .1)  $q \notin F$ , and
  - .2)  $\forall r \in R_s[q] \exists a \in A(r), -\Delta a \in P[1..i]$ .

**Fig. 1.** Inference rules for  $+\Delta$  and  $-\Delta$

antecedent, or body, of rule  $r$  and  $P(i+1)$  ( $P[1..i]$ ) refers to the  $i+1$ 'th tagged literal in the proof  $P$  (the initial segment of  $P$  of length  $i$ ). The  $+\Delta$  inference rule is essentially *modus ponens* while the  $-\Delta$  inference rule is the strong negation [4] of the  $+\Delta$  rule. The  $\Delta$  inference rules are sometimes referred to as the monotonic part of the defeasible logic, since defeasibility is not involved.

Notice that if  $D$  contains facts  $p$  and  $\neg p$  then we have both  $+\Delta p$  and  $+\Delta \neg p$  as consequences of  $D$ .

The inference rules for  $\partial$  are substantially more complicated (see Figure 2). They must take into account  $\Delta$  inferences, all three kinds of rules and the superiority relation. Roughly speaking,  $+\partial q$  can be derived if either  $p$  can be proved monotonically, or (1) there is a strict or defeasible

- $+\partial$ ) If  $P(i+1) = +\partial q$  then either
- .1)  $+\Delta q \in P[1..i]$ ; or
  - .2) The following three conditions all hold.
    - .1)  $\exists r \in R_{sd}[q] \forall a \in A(r), +\partial a \in P[1..i]$ , and
    - .2)  $-\Delta \sim q \in P[1..i]$ , and
    - .3)  $\forall s \in R[\sim q]$  either
      - .1)  $\exists a \in A(s), -\partial a \in P[1..i]$ ; or
      - .2)  $\exists t \in R_{sd}[q]$  such that
        - .1)  $\forall a \in A(t), +\partial a \in P[1..i]$ , and
        - .2)  $t > s$ .
- $-\partial$ ) If  $P(i+1) = -\partial q$  then
- .1)  $-\Delta q \in P[1..i]$ , and
  - .2) either
    - .1)  $\forall r \in R_{sd}[q] \exists a \in A(r), -\partial a \in P[1..i]$ ; or
    - .2)  $+\Delta \sim q \in P[1..i]$ ; or
    - .3)  $\exists s \in R[\sim q]$  such that
      - .1)  $\forall a \in A(s), +\partial a \in P[1..i]$ , and
      - .2)  $\forall t \in R_{sd}[q]$  either
        - .1)  $\exists a \in A(t), -\partial a \in P[1..i]$ ; or
        - .2)  $\text{not}(t > s)$ .

**Fig. 2.** Inference rules for  $+\partial$  and  $-\partial$

rule for  $q$ , the body of which can be proved defeasibly; (2)  $\sim q$  cannot be proved monotonically; and (3) every rule for  $\sim q$  that is applicable is over-ruled by an applicable rule for  $q$ . This latter behaviour (3) is called team defeat: the rules for  $q$ , as a team, over-rule the rules for  $\sim q$ , even though possibly no individual rule for  $q$  over-rules all its competitors. For the simpler inferences where we require a single rule to over-ride all its competitors, the symbol  $\partial^*$  is used. Again, failure to prove  $(-\partial)$  is the strong negation of  $+\partial$ .

Defeasible logics other than **DL** use different tags and inference rules. We will later refer to tags  $\delta$  and  $f$ , whose inference rules are defined in [17, 7].

## 2.2 Non-monotonic Inheritance Networks

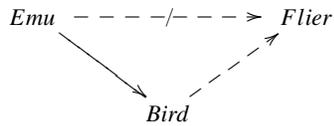
Inheritance hierarchies are used to represent a taxonomic-style division of classes into subclasses. Thus they can describe a situation where, say, the class of emus inherits properties (such as having feathers) from the class of birds. Inheritance hierarchies can be represented by simple implications employing unary predicates. For example  $emu(X) \rightarrow bird(X)$  represents that emu is a sub-class of bird.

However, emus are exceptional in the class of birds and to represent such situations where properties are generally – but not universally – inherited by subclasses, a weaker form of inheritance that admits exceptions is added.

An inheritance network can be described by a directed acyclic graph. Vertices of the graph represent properties (or classes) such as *emu* or *bird*. A solid arrow represents a strict (or definite, or certain) inference while a dashed arrow represents a defeasible inference. An arrow with a slash represents an exception: an inference of the negation of the target. For example, Figure 3 presents the well-known Tweety problem and an inheritance network representing it.

Inheritance hierarchies can be represented by simple implications employing unary predicates. For example  $emu(X) \rightarrow bird(X)$  represents that emu is a sub-class of bird. For weaker inheritance statements we can use a defeasible implication. Thus  $bird(X) \Rightarrow flier(X)$  represents that birds are generally fliers. Finally, exceptions can be expressed as implications to a negated literal. For example,  $emu(X) \Rightarrow \neg flier(X)$ . Facts state

All emus are birds  
 Usually, birds fly  
 Usually, emus do not fly



**Fig. 3.** Inheritance network for the Tweety problem.

that constants are members of classes. For example, the fact  $emu(tweety)$  expresses that Tweety is a member of the emu class.

Non-monotonic inheritance networks are one of the earliest formalisms for non-monotonic reasoning, and perhaps the earliest formalism involving defeasible reasoning. Early in the development of defeasible logic it was recognised that defeasible logic could be considered a generalization of non-monotonic inheritance networks [6]. Syntactically, inheritance networks are a very small subset of defeasible theories: only unary predicates are admitted, only one variable can be used in a rule, bodies of rules must contain exactly one atomic formula, and heads must contain exactly one literal.

As mentioned in the Introduction, there are many different semantics for non-monotonic inheritance networks. In the next section we will see a semantics for these networks inspired by empirical study of human reasoning.

### 3 Ford's Logic

Ford developed the logic **LS** from observation of human reasoning on inheritance problems and Gilio's analysis [14] of a probabilistic interpretation of system **P** [21].

Ford's logic is partly based on the recognition that chains of reasoning involving both strict and defeasible arrows may have different strengths. Consider the chain (1) below. If a typical  $A$  is a  $B$ , and every  $B$  is a  $C$  then it must be that all typical  $A$ 's are  $C$ 's, so we might be justified in drawing a defeasible arrow directly from  $A$  to  $C$ . In (2), however, every  $A$  is a  $B$  and a typical  $B$  is a  $C$  but we have no assurance that any  $A$  is a typical  $B$ . Thus following the chain from  $A$  to  $C$  makes an implicit and unsupported assumption that a typical  $A$  is typically a typical  $B$ . There is a similar problem in chain (3).

$$(1) \quad A \dashv\vdash B \longrightarrow C$$

$$(2) \quad A \longrightarrow B \dashv\vdash C$$

$$(3) \quad A \dashv\vdash B \dashv\vdash C$$

This distinction between (1) and (2–3) was used by some of Ford’s experimental subjects [13, 11] and is also implicitly recognised in the research literature in the study of the Pennsylvania Dutch problem [19]. In terms of the inference rules discussed in [21], Ford identifies the weakness of (2) and (3) with the weakness of the Monotonicity and Transitivity rules.

**LS** is based on three tiers of defeasible inference, roughly characterized as follows [10]:

$\alpha \vdash_1 \beta$  *more than half of the  $\alpha$  are  $\beta$*

$\alpha \vdash_2 \beta$  *some  $\alpha$  are  $\beta$*

$\alpha \vdash_3 \beta$  *a relationship has been derived, but it is possible no  $\alpha$  are  $\beta$*

The index on  $\vdash$  represents the “logical strength” of the relation, where smaller means stronger.

Thus, if we associate typicality/normality with more than half a population (which is in line with English usage), in chain (1) we feel justified in claiming  $A \vdash_1 C$  while in chains (2) and (3) we can only claim  $A \vdash_3 C$ .

The intermediate tier of inference  $\vdash_2$  is needed once conjunction is introduced. For example, it may be that  $A$ ’s are usually  $B$ ’s and  $A$ ’s are usually  $C$ ’s, but we cannot conclude that  $A$ ’s usually satisfy  $B \wedge C$ . This is because it might be that the population of  $A$ ’s is largely polarized into individuals that are  $B$ ’s but not  $C$ ’s and individuals that are  $C$ ’s but not  $B$ ’s, with only a small proportion satisfying  $B \wedge C$ . On the other hand, if more than 75% of  $A$ ’s are  $B$ ’s and 75% of  $A$ ’s are  $C$ ’s then more than half of  $A$ ’s satisfy  $B \wedge C$ . Thus the use of the intermediate tier allows the logic to recognise the greater strength of the inference of  $B \wedge C$  from  $A$  than the inference in (2) or (3) above.

The inference rules for **LS** are given in Figure 4. CM refers to Cautious Monotonicity. Throughout Ford’s presentation of the logic, a collection of inheritance statements – both strict and defeasible – is assumed to

$$\begin{array}{l}
\text{Right Weakening} \quad \frac{\models \alpha \rightarrow \beta \quad \gamma \vdash_n \alpha}{\gamma \vdash_n \beta} \quad \text{S} \quad \frac{\alpha \wedge \beta \vdash_n \gamma}{\alpha \vdash_n \beta \rightarrow \gamma} \\
\\
\text{Left Logical Equivalence} \quad \frac{\models \alpha \leftrightarrow \beta \quad \alpha \vdash_n \gamma}{\beta \vdash_n \gamma} \\
\\
\text{And (1)} \quad \frac{\alpha \vdash_1 \beta \quad \alpha \vdash_1 \gamma}{\alpha \vdash_2 \beta \wedge \gamma} \quad \text{CM (1)} \quad \frac{\alpha \vdash_1 \beta \quad \alpha \vdash_1 \gamma}{\alpha \wedge \beta \vdash_2 \gamma} \\
\\
\text{And (2)} \quad \frac{\alpha \vdash_n \beta \quad \alpha \vdash_m \gamma}{\alpha \vdash_3 \beta \wedge \gamma} \quad \text{CM (2)} \quad \frac{\alpha \vdash_m \beta \quad \alpha \vdash_n \gamma}{\alpha \wedge \beta \vdash_3 \gamma} \\
\\
\text{Cut (1)} \quad \frac{\alpha \wedge \beta \vdash_m \gamma \quad \alpha \vdash_n \beta}{\alpha \vdash_2 \gamma} \quad \text{Or (1)} \quad \frac{\alpha \vdash_m \gamma \quad \beta \vdash_n \gamma}{\alpha \vee \beta \vdash_2 \gamma} \\
\\
\text{Cut (2)} \quad \frac{\alpha \wedge \beta \vdash_m \gamma \quad \alpha \vdash_n \beta}{\alpha \vdash_3 \gamma} \quad \text{Or (2)} \quad \frac{\alpha \vdash_m \gamma \quad \beta \vdash_n \gamma}{\alpha \vee \beta \vdash_3 \gamma} \\
\\
\text{Monotonicity} \quad \frac{\models \alpha \rightarrow \beta \quad \beta \vdash_n \gamma}{\alpha \vdash_3 \gamma} \quad \text{Transitivity} \quad \frac{\alpha \vdash_m \beta \quad \beta \vdash_n \gamma}{\alpha \vdash_3 \gamma}
\end{array}$$

Some inference rules have side-conditions: And (2) and CM (2) require  $n \geq 2$  or  $m \geq 2$ ; Cut (1) and Or (1) require  $1 \leq n \leq 2$  and  $1 \leq m \leq 2$ ; and Cut (2) and Or (2) require  $n = 3$  or  $m = 3$ .

**Fig. 4.** Inference rules for **LS**

exist as a parameter to the inference. Presumably, a statement  $\models \alpha \rightarrow \beta$  refers to a consequence of the strict inheritance statements. Similarly, we presume that for every defeasible inheritance of  $\alpha$  from  $\beta$  we can infer  $\beta \vdash_1 \alpha$ .

Ford leaves implicit inference rules that allow inferring  $\alpha \vdash_{m+1} \beta$  from  $\alpha \vdash_m \beta$  for  $m = 1, 2$ . Without these, inference relation  $\vdash_2$  is not contained in  $\vdash_3$  as a result of (say) the restrictions on And(2). But it is clear that this presentation of inference rules like And is used to focus on the *strongest* inference that is permitted by the logic, and is not intended

to exclude concluding  $\alpha \sim_3 \beta \wedge \gamma$  from  $\alpha \sim_1 \beta$  and  $\alpha \sim_1 \gamma$ . Similarly, a Reflexivity axiom  $\alpha \sim_m \alpha$  could be added to **LS**.

When conflicting inferences can be derived in **LS**, say both  $\alpha \sim_i \beta$  and  $\alpha \sim_j \neg\beta$ , the conclusion that is drawn is based on the logical strengths of the two statements. If  $i < j$  then we conclude  $\alpha \sim \beta$  and if  $i > j$  then we conclude  $\alpha \sim \neg\beta$ . If  $i = j$  we cannot draw any conclusion.

## 4 The Logic FDL

The defeasible logic derived from Ford's logic will be called **FDL**. There are several points of difference between Ford's logic **LS** and existing defeasible logics:

1. **LS** admits a broader syntax than defeasible logics, including the connectives  $\vee$  and  $\rightarrow$ .
2. **LS** infers defeasible implications whereas defeasible logics infer literals.
3. Defeasible logics may express priorities among rules using the superiority relation, which is not available in **LS**.
4. **LS** supports three tiers of defeasible inference, whereas most defeasible logics support only one.
5. **LS** takes a meta-theoretic approach to the resolution of conflicting conclusions: if we can derive  $\alpha \sim_1 \beta$  and  $\alpha \sim_2 \neg\beta$  then we conclude  $\beta$  defeasibly holds, based on the relative strength of the two inference relations. In contrast, in every defeasible logic investigated thus far, the resolution of conflicts is embedded in the inference rules. For example, in the inference rule for  $+\partial$ , condition 2.2 ensures that there is not a conflicting conclusion of greater strength.
6. **LS** does not express statements about literals that are not derivable, whereas defeasible logics provide inference rules for deriving such statements.

In **FDL** we will restrict the syntax of **LS** to the defeasible logic style to address the first two points. **FDL** will not use the superiority relation, but new tags  $+\partial_1$ ,  $+\partial_2$ , and  $+\partial_3$  are introduced in **FDL** to reflect the three tiers in **LS**. In place of the meta-logical balancing of logical strength in **LS** we will incorporate the resolution of conflicts within the inference rules, in the defeasible logic style. The end result is a defeasible logic that

is guided by the principles of human commonsense reasoning identified by Ford but does not have the syntactic breadth of **LS**.

We now turn to the definition of **FDL** and motivation for some of the technical decisions.

Ford's logic, and the distinction between the three inference relations is partly based on the relative weakness of some paths, compared to others. This weakness relies on the loss of the presumption of typicality after the application of a rule. For example, although Tweety is an emu, and all emus are birds, the presumption that Tweety is a typical emu does not lead to a presumption that Tweety is a typical bird. Even if we were given, separately, the fact that Tweety is a bird, which would carry the presumption that Tweety is typical of birds, the existence of information that Tweety belongs to a subclass of birds should lead us at least to question – and perhaps to invalidate – the presumption.

Taking the latter approach, we must distinguish between facts that carry with them a presumption of typicality and facts that carry no such presumption. This distinction will be reflected in the tags we use. For facts that may or may not carry a presumption, we can use the  $\Delta$  tags, while for facts that carry a presumption we introduce a new tag  $\Phi$ . The inference rules for  $\Phi$  are in Figure 5.

- $+\Phi$ ) If  $P(i+1) = +\Phi q$  then either
  - .1)  $q \in F$ ; and
  - .2)  $\forall r \in R_s[q] \exists a \in A(r), -\Delta a \in P[1..i]$ .
- $-\Phi$ ) If  $P(i+1) = -\Phi q$  then
  - .1)  $q \notin F$ , or
  - .2)  $\exists r \in R_s[q] \forall a \in A(r), +\Delta a \in P[1..i]$ .

**Fig. 5.** Inference rules for  $+\Phi$  and  $-\Phi$

It is worthwhile noting that, as a result of this distinction, facts are not equivalent to strict rules with no antecedent, as is true in **DL**. Hence, a simplification of [1] which eliminates facts in **DL** is not valid for **FDL**.

To make matters simpler we will separate out inferring the status of conjunctions from inference involved in the application of a rule. We introduce tags on conjunctions for this purpose. Let  $p$  and  $q$  range over conjunctions of literals (including the trivial conjunction consisting of a

single literal). The positive inference rules for conjunction are as follows.

$$\begin{aligned}
&\text{If } P(i+1) = +\Delta(p \wedge q) \text{ then } +\Delta p \in P[1..i] \text{ and } +\Delta q \in P[1..i] \\
&\text{If } P(i+1) = +\partial_1(p \wedge q) \text{ then } +\Delta(p \wedge q) \in P[1..i] \\
&\text{If } P(i+1) = +\partial_2(p \wedge q) \text{ then } +\partial_1(p \wedge q) \in P[1..i] \text{ or} \\
&\quad (+\partial_1 p \in P[1..i] \text{ and } +\partial_1 q \in P[1..i]) \\
&\text{If } P(i+1) = +\partial_3(p \wedge q) \text{ then } +\partial_2(p \wedge q) \in P[1..i] \text{ or} \\
&\quad (+\partial_3 p \in P[1..i] \text{ and } +\partial_3 q \in P[1..i])
\end{aligned}$$

The negative inference rules are the strong negations of these rules.

The inference rules explicitly allow a conjunction to receive a tag if it has received a stronger tag. No conjunctions are tagged with  $+\Phi$ ; this follows from the intended meaning of  $\Phi$  to reflect that a fact *emu(tweety)* expresses a presumption that Tweety is a typical emu. The first inference rule is simply a reflection of the conjunction implicitly used in the inference rule for  $+\Delta$ . The third inference rule reflects And(1) while the fourth reflects And(2). Following the And rules, a conjunction can have tag  $+\partial_1$  only if the conjunction definitely holds (i.e. has tag  $+\Delta$ ).

If we view conjunction as an operation on tags, it is commutative but not associative. For example, if  $+\Delta p$ ,  $+\Delta q$ , and  $+\partial_1 r$  then we can infer  $+\partial_2((p \wedge q) \wedge r)$  but only infer  $+\partial_3(p \wedge (q \wedge r))$ . To extend conjunction to sets of literals we define the tag of the conjunction to be the strongest tag achievable by any grouping and ordering of the literals. This is equivalent to requiring that all  $\Delta$  tagged literals are grouped together before applying binary conjunctions.

We now turn to the inference rules for tagged literals based on the application of rules. These rules are presented in Figure 6. As with the other tags we have used, the inference rule for  $-\partial_i$  is the strong negation of the inference rules for  $+\partial_i$ , for each  $i$ .

There are several elements of the rules of interest. The conditions 1 in the inference rules ensure that there is a hierarchy of inference strength, from  $\Delta$  to  $\partial_3$ . The conditions 2.1 for strict rules reflect Right Weakening in **LS**. Condition 2.1 for  $\partial_3$  reflects Monotonicity and Transitivity.

In drawing conclusions with Ford's logic, a stronger inference overrides a conflicting weaker inference, although this occurs at the meta level. A similar behaviour is ensured in **FDL** from within the logic. Conditions 2.2 of the inference rules specify that to infer  $+\partial_i q$  we must prove that  $\sim q$  cannot be proved at a greater strength. Similarly, the combination of conditions 2.1 and 2.3 (and 2.4 for  $\partial_1$ ) ensures that when we

- $+ \partial_1$ ) If  $P(i+1) = + \partial_1 q$  then either
- .1)  $+ \Delta q \in P[1..i]$ ; or
  - .2) The following four conditions all hold.
    - .1)  $\exists r \in R_s[q], + \partial_1 A(r) \in P[1..i]$  or  
 $\exists r \in R_d[q], + \Phi A(r) \in P[1..i]$ , and
    - .2)  $- \Delta \sim q \in P[1..i]$ , and
    - .3)  $\forall s \in R_s[\sim q], - \partial_1 A(s) \in P[1..i]$ , and
    - .4)  $\forall s \in R_{dd}[\sim q], - \Phi A(s) \in P[1..i]$
- $+ \partial_2$ ) If  $P(i+1) = + \partial_2 q$  then either
- .1)  $+ \partial_1 q \in P[1..i]$ ; or
  - .2) The following three conditions all hold.
    - .1)  $\exists r \in R_s[q], + \partial_2 A(r) \in P[1..i]$ , and
    - .2)  $- \partial_1 \sim q \in P[1..i]$ , and
    - .3)  $\forall s \in R_s[\sim q], - \partial_2 A(s) \in P[1..i]$
    - .4)  $\forall s \in R_{dd}[\sim q], - \Phi A(s) \in P[1..i]$
- $+ \partial_3$ ) If  $P(i+1) = + \partial_3 q$  then either
- .1)  $+ \partial_2 q \in P[1..i]$ ; or
  - .2) The following three conditions all hold.
    - .1)  $\exists r \in R_{sd}[q], + \partial_3 A(r) \in P[1..i]$ , and
    - .2)  $- \partial_2 \sim q \in P[1..i]$ , and
    - .3)  $\forall s \in R[\sim q], - \partial_3 A(s) \in P[1..i]$

**Fig. 6.** Inference rules for  $+ \partial_i$  for  $i = 1, 2, 3$

have competing inferences of equal strength no positive conclusion can be drawn.

For inheritance networks we can view the tags as encoding information about paths in the inheritance network.  $\Phi$  represents the empty path,  $\Delta$  paths consisting only of strict rules,  $\partial_1$  paths of the form  $\Rightarrow \rightarrow^*$ , and  $\partial_3$  represents all paths. The motivation for introducing  $\vdash_2$  in **LS** was not related to paths and so its counterpart  $\partial_2$  in **FDL** encodes no larger class of paths than  $\partial_1$ .

## 5 Properties of FDL

We now turn to investigate the properties of this logic. The fact that **FDL** can be seen as an instance of the framework for defeasible logics [4] greatly simplifies the establishment of these properties. The first property concerns the efficiency of inference in the logic. In line with other defeasible logics [23, 7], inference can be performed in linear time. This is in contrast to many other non-monotonic logics, for which inference is NP-hard.

**Proposition 1.** *The set of all literals that can be inferred from a propositional **FDL** theory  $D$  can be computed in  $O(N)$  time, where  $N$  is the size of  $D$ .*

**FDL** is amenable to efficient implementation in the same way that other defeasible logics are [3, 23, 24].

Coherence of a defeasible logic refers to the property that tagged literals obtained by applying complementary tags to the same literal cannot both be inferred.

**Definition 1.** *A defeasible logic is coherent if, for every defeasible theory  $D$  in the logic, every tag  $d$ , and every literal  $q$ , we do not have both  $D \vdash +dq$  and  $D \vdash -dq$ .*

It is this property that supports the intuition that  $+d$  represents a form of provability while  $-d$  represents finite failure to prove: we can never both prove and be unable to prove a proposition. As might be expected, **FDL** enjoys this property.

**Proposition 2.** ***FDL** is coherent.*

The monotonic part of a defeasible theory allows the inference of both a proposition and its negation. Thus defeasible logics may be inconsistent in the usual sense, independent of the defeasible inferences. Since the defeasible inferences are the main focus of defeasible logics, consistency for defeasible theories refers to a guarantee that the only contradictory information that is derived from a theory is already a consequence of the monotonic part of the theory alone.

**Definition 2.** *A defeasible logic is relatively consistent if, for every defeasible theory  $D$  in the logic, for every tag  $d$ , and every proposition  $q$ , we do not have  $D \vdash +dq$  and  $D \vdash +d\neg q$  unless  $D \vdash +\Delta q$  and  $D \vdash +\Delta \neg q$ .*

Again, this is a property that holds for **FDL**. In this sense, the logic is paraconsistent: beyond conflicting strict statements, no inconsistent inferences are made.

**Proposition 3.** *FDL is relatively consistent.*

Decisiveness insists that the proof theory determines the proof status of every tagged literal. It is a form of inverse of coherence. A logic is *decisive* for a defeasible theory  $D$  if, every tag  $d$ , and every literal  $q$ , either  $D \vdash +dq$  or  $D \vdash -dq$ . A propositional defeasible theory  $D$  is acyclic if its dependency graph is acyclic. All non-monotonic inheritance networks form acyclic defeasible theories. The following result shows that, in particular, **FDL** is decisive on non-monotonic inheritance networks.

**Proposition 4.** *If the defeasible theory  $D$  is acyclic then **FDL** is decisive for  $D$ .*

Propositions 1–4 identify properties that **FDL** has in common with many other defeasible logics. We now look at inference strength, where we will see a difference between **FDL** and **DL**.

When comparing inference strength we use the following notation. Let  $d_1$  and  $d_2$  be tags. We write  $d_1 \subseteq d_2$  to say that, for any defeasible theory  $D$ ,  $\{p \mid D \vdash +d_1p\} \subseteq \{p \mid D \vdash +d_2p\}$  and  $\{p \mid D \vdash -d_1p\} \supseteq \{p \mid D \vdash -d_2p\}$ . That is, we can derive  $p$  with  $d_2$  any time we can derive  $p$  with  $d_1$  and, conversely, we recognise that we cannot derive  $p$  with  $d_1$  any time we recognise that we cannot derive  $p$  with  $d_2$ .

Almost immediately from the definitions we have a hierarchy of inference strength among the tags of **FDL**

**Proposition 5.**

$$\Phi \subseteq \Delta \subseteq \partial_1 \subseteq \partial_2 \subseteq \partial_3$$

It is easy to see that these containments are *strict*, that is, for every  $d \subseteq d'$ , there is a defeasible theory  $D$  and literal  $q$  such that  $D \vdash +dq$  but  $D \not\vdash +d'q$  and/or  $D \vdash -d'q$  but  $D \not\vdash -dq$ .

**FDL** does not involve a superiority relation. In comparing with other defeasible logics we consider only defeasible theories without a superiority relation. In that case, the logics with and without team defeat are equivalent. That is,  $\partial = \partial^*$  and similarly  $\delta = \delta^*$  and  $f = f^*$ . We now see that **FDL** and **DL** are incomparable in inference strength.

**Proposition 6.**  $\partial_1, \partial_2,$  and  $\partial_3$  are incomparable to  $\partial$ . That is,  $\partial_i \not\subseteq \partial$  and  $\partial \not\subseteq \partial_i$ , for  $i = 1, \dots, 3$ ,

This result can be established using one simple defeasible theory. We will see that  $\partial_1 \not\subseteq \partial$  and  $\partial \not\subseteq \partial_3$ . It follows immediately that  $\partial_2 \not\subseteq \partial$  and  $\partial_3 \not\subseteq \partial$  and, similarly,  $\partial \not\subseteq \partial_1$  and  $\partial \not\subseteq \partial_2$ .

Consider a defeasible theory  $D$  consisting of facts  $p, r$  and  $s$  and rules

$$\begin{aligned} p &\Rightarrow q \\ r &\Rightarrow \neg q \\ s &\rightarrow r \\ q &\Rightarrow t \\ r &\Rightarrow \neg t \end{aligned}$$

Then  $+\Delta p$  and  $+\Delta r$ , and  $-\Delta t$ . Consequently, in **DL** we have equal support for  $q$  and  $\neg q$  and hence we derive  $-\partial q$ . As a result, we can derive  $+\partial \neg t$ .

On the other hand, in **FDL** we have  $+\Phi p$  and  $+\Phi s$  but  $-\Phi r$ . Hence, condition 2.1 for  $+\partial_1 q$  and condition 2.4 are both satisfied. Furthermore, there is no strict rule for  $\neg q$ , so conditions 2.2 and 2.3 are satisfied. Thus we conclude  $+\partial_1 q$ . This establishes that  $\partial_1 \not\subseteq \partial$ .

Considering the status of  $\neg t$  under **FDL**, we see that  $-\partial_1 \neg t$  because condition 2.1 is satisfied. Consequently also  $-\partial_2 \neg t$  by condition 2.1. Finally, by condition 2.3 of  $\partial_3$  we must conclude  $-\partial_3 \neg t$ , since the antecedent of the rule for  $t$  is  $q$  and we know  $+\partial_1 q$ , and hence  $+\partial_3 q$ . This establishes that  $\partial \not\subseteq \partial_3$ .

Combining this result with results of [7] and some other results, we see the relationship between inference in **FDL** and inference in other defeasible logics.

**Proposition 7.** Consider a defeasible theory  $D$  containing no superiority statements. Then the following containments hold.

$$\begin{array}{ccc} & \partial_1 \subseteq \partial_2 \subseteq \partial_3 & \\ & \subsetneq & \subsetneq \\ \Phi \subseteq \Delta & & \int \\ & \subsetneq & \subsetneq \\ & \delta \subseteq \partial & \end{array}$$

All of the containments in the above proposition are strict. Only the status of  $\delta \subseteq \partial_3$  is not known.

## 6 Conclusion

We have seen a logic **FDL** that provides a model of human “common sense” reasoning. The logic is less expressive than Ford’s System **LS** in that it does not support connectives  $\vee$  and  $\rightarrow$ , but it has a low complexity and is amenable to efficient implementation. It satisfies several desirable properties and we have seen its relationship to other defeasible logics. **FDL** in general makes different inferences than **DL** and other defeasible logics. There are even greater differences with logics and argumentation systems based on stable semantics, which tend to make more positive inferences than **DL**.

A computational model of human “common sense” reasoning is useful for any reasoning system that interacts with people. For these systems there is a danger that there is a mismatch between the logic it employs and the logic that people use or would accept as valid reasoning. For example, a formalization of regulations [5], business rules [18], or contracts [27, 15] as defeasible rules might lead to differences in interpretation between the people who formulate the rules (and the people who must follow the rules) and systems that enforce the rules. Similarly, if a software agent’s desired behaviour is described with defeasible rules by a naive user [8] then there is a danger that the implemented behaviour differs from the expected behaviour. Finally, for software agents that interact directly with people, an accurate model of human reasoning is useful, simply to understand the interaction properly.

There remains much more to do if we are to harmonise human and software reasoning. Although the design of **FDL** builds on **LS**, the formal relationship between the two logics has not been addressed. Clarifying this relationship will be necessary. However, there may be better models of human commonsense reasoning than **LS** and **FDL** and there is great scope for experimental and theoretical work on this subject.

### Acknowledgments

My thanks to Marilyn Ford for discussions and several drafts of her work developing the logic **LS**, and to Grigoris Antoniou, David Billington, and Guido Governatori for their collaboration on defeasible logic over many years. And thanks to Guido for his comments on a rough draft.

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Econ-

omy and the Australian Research Council through the ICT Centre of Excellence program.

## References

1. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic* 2 (2001): 255–287
2. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. Embedding Defeasible Logic into Logic Programming. *Theory and Practice of Logic Programming* 6 (2006): 703–735.
3. G. Antoniou, D. Billington, G. Governatori, M.J. Maher, and A. Rock. A family of defeasible reasoning logics and its implementation. *Proc. 14th European Conference on Artificial Intelligence (ECAI 2000)*, 459–463, IOS Press. 2000.
4. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. A Flexible Framework for Defeasible Logics. *Proc. National Conference on Artificial Intelligence (AAAI-00)*, 405–410, 2000.
5. G. Antoniou, D. Billington and M.J. Maher. On the analysis of regulations using defeasible rules. In *Proc. 32nd Hawaii International Conference on Systems Science*, 1999.
6. D. Billington, K. de Coster and D. Nute. A Modular Translation from Defeasible Nets to Defeasible Logic. *Journal of Experimental and Theoretical Artificial Intelligence* 2 (1990): 151–177.
7. D. Billington, G. Antoniou, G. Governatori and M.J. Maher. An Inclusion Theorem for Defeasible Logics. *ACM Transactions on Computational Logic*, to appear.
8. M. Dumas, G. Governatori, A. ter Hofstede and P. Oaks. A formal approach to negotiating agents development. *Electronic Commerce Research and Applications* 1 (2002): 193–207.
9. J.St.B.T. Evans. *Bias in Human Reasoning*. Erlbaum, 1989.
10. M.Ford. System LS: A three-tiered nonmonotonic reasoning system. *Computational Intelligence* 20, 1, 89–107, 2004.
11. M.Ford. Human nonmonotonic reasoning: The importance of seeing the logical strength of arguments. *Synthese* 146, 1&2, 71–92, 2005.
12. M.Ford. On using human nonmonotonic reasoning to inform artificial systems. *Psychologica Belgica*, 45(1), 57–70, 2005.
13. M. Ford and D. Billington. Strategies in human commonsense reasoning. *Computational Intelligence* 16, 446–468, 2000.
14. A. Gilio. Probabilistic Reasoning Under Coherence in System P. *Annals of Mathematics and Artificial Intelligence* 34(1-3): 5–34, 2002.
15. G. Governatori. Representing business contracts in RuleML. *International Journal of Cooperative Information Systems* 14 (2005): 181–216
16. G. Governatori and M.J. Maher. An Argumentation-Theoretic Characterization of Defeasible Logic. *Proc. 14th European Conference on Artificial Intelligence (ECAI 2000)*, 469–474, IOS Press. 2000.
17. G. Governatori, M.J. Maher, G. Antoniou, and D. Billington. Argumentation Semantics for Defeasible Logics. *Journal of Logic and Computation* 14 (2004): 675–702.
18. B.N. Groszof, Y. Labrou and H.Y. Chan. A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML. In *Proc. 1st ACM Conference on Electronic Commerce (EC-99)*, ACM Press 1999.
19. J.F. Horty and R.H. Thomason. Mixing Strict and Defeasible Inheritance. *Proc. National Conference on Artificial Intelligence (AAAI-88)*, 427–432, 1988.
20. P.N. Johnson-Laird and R.M.J. Byrne. *Deduction*. Erlbaum, 1991.

21. S. Kraus, D. Lehmann and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44 (1990): 167–207.
22. M.J. Maher. A Denotational Semantics for Defeasible Logic. *Proc. First International Conference on Computational Logic*, LNAI 1861, Springer, 2000, 209–222.
23. M.J. Maher. Propositional Defeasible Logic has Linear Complexity. *Theory and Practice of Logic Programming*, 1 (2001): 691–711.
24. M.J. Maher, A. Rock, G. Antoniou, D. Billington, and T. Miller. Efficient Defeasible Reasoning Systems. *International Journal on Artificial Intelligence Tools* 10(4): 483–501, 2001.
25. M.J. Maher. A Model-Theoretic Semantics for Defeasible Logic. *Proc. Workshop on Paraconsistent Computational Logic*, 67–80, 2002.
26. D. Nute. 1994. Defeasible Logic. In D.M. Gabbay, C.J. Hogger and J.A. Robinson (eds.): *Handbook of Logic in Artificial Intelligence and Logic Programming Vol. 3*, Oxford University Press, 353–395.
27. D.M. Reeves, B.N. Grosz, M.P. Wellman, and H.Y. Chan. Towards a Declarative Language for Negotiating Executable Contracts, *Proc. AAAI-99 Workshop on Artificial Intelligence in Electronic Commerce (AIEC-99)*, AAAI Press / MIT Press, 1999.
28. D.D. Touretzky, J.F. Horty and R.H. Thomason. 1987. A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems. *Proc. 10th International Joint Conference on Artificial Intelligence (IJCAI-87)*, 476–482, Morgan Kaufmann, 1987.