An Introduction to Clausal Defeasible Logic

Clausal Defeasible Logic reasons with factual information, like “Whales are mammals.”, and plausible information, like “Mammals usually do not lay eggs.”; recall that the platypus and echidna of Australia are mammals that do lay eggs. The factual and plausible information is represented by strict rules, defeasible rules, warning rules, and a priority relation on the rules. All rules have the form "finite-set-of-literals arrow literal".

Strict rules, for example $A \rightarrow l$, represent the aspects of a situation which are certain. If all the literals in A are proved then l can be deduced, no matter what the evidence against l is. So strict rules behave like material (ordinary) implication. For example, “Emus are birds.” is represented as the strict rule (schema) $emu(x) \rightarrow bird(x)$, and its contrapositive $\sim bird(x) \rightarrow \sim emu(x)$. (If the antecedent of a rule is a singleton set we often omit the set braces.)

Defeasible rules, for example $A \Rightarrow l$, represent some of the aspects of a situation which are plausible. If all the literals in A are proved then l can be deduced provided that all the evidence against l has been nullified. So we take $A \Rightarrow l$ to mean that, in the absence of evidence against l, A is sufficient evidence for concluding l. For example, “Birds usually fly.” is represented by $bird(x) \Rightarrow fly(x)$. The idea is that if we know that something is a bird, then we may conclude that it flies, unless there is other evidence suggesting that it may not fly.

A warning rule, for example $A \rightsquigarrow \sim l$, is evidence against l. In the absence of other rules, $A \rightsquigarrow \sim l$ means that if A is not disproved then it is too risky to conclude l. Warning rules are used to prevent conclusions which would be too risky. For example, “Sick birds might not fly.” is represented by $\{sick(x), bird(x)\} \rightsquigarrow \sim fly(x)$. The idea is that a bird being sick is not sufficient evidence to conclude that it does not fly; it is only evidence against the conclusion that it usually flies. Another use for warning rules is to cut chains of defeasible rules which are too long. For instance, given $a \Rightarrow b$ and $b \Rightarrow c$ it may be too risky to conclude that c holds given that a holds. In which case we could add the warning rule $a \rightsquigarrow \sim c$. The point is that adding $a \Rightarrow \sim c$ instead of $a \rightsquigarrow \sim c$ would be wrong, because accepting a is not a reason for accepting $\sim c$, indeed it is a weak reason for accepting c.

The priority relation, >, on the set of rules allows the representation of preferences among rules. The priority relation must be acyclic. For example consider the following (Tweety triangle) situation.

$\{\} \rightarrow quail(Quin)$	[Quin is a quail.]
$quail(x) \rightarrow bird(x)$	[Quails are birds.]
$\sim bird(x) \rightarrow \sim quail(x)$	[Contrapositive of Quails are birds.]
$R1$: $bird(x) \Rightarrow fly(x)$	[Birds usually fly.]
$R2$: $quail(x) \Rightarrow \sim fly(x)$	[Quails usually do not fly.]

We want to conclude that usually Quin does not fly. But this can only be done if we prefer $R2$ to $R1$, hence we define $R2 > R1$.

A clausal defeasible theory, $\Theta = (R, >)$, consists of a set of rules R and a priority relation > on R. The task of proving a formula is done by a recursive function P called the proof function of Θ. The input to P includes the proof algorithm to be used and the formula to be proved. The output of P is either +1, 0, or –1. Essentially we have a three valued logic in which +1 denotes proved, 0 denotes loops, and –1 denotes that there is a demonstration that the formula is not provable and does not generate a loop.

Most non-monmotonic logics do not distinguish between formulas proved using only factual information and those proved using plausible information. Clausal Defeasible Logic does. Indeed Clausal Defeasible Logic distinguishes between formulas proved with a variety of different proof algorithms. The μ proof algorithm only uses factual information, and Clausal Defeasible Logic restricted to just the μ algorithm is essentially classical propositional logic. The π proof algorithm propagates ambiguity, and the β proof algorithm blocks ambiguity. Every formula which is provable using μ can be proved using π; and every formula which is provable using π can be proved using β. So both intuitions about the blocking or propagating of ambiguity are catered for. Moreover we see how they are related.

Clausal Defeasible Logic uses team defeat, detects all loops in a finite number of steps, and uses failure-by-looping.

An earlier version of Clausal Defeasible Logic has been used in robot dogs which play soccer, and in a robot designed to raise an alerm if an elderly person needs help.