Is Defeasible Logic Applicable?*

Guido Governatori, Arthur H.M. ter Hofstede and Phillipa Oaks
Centre for Cooperative Information Systems
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia
email: {governat,arthur,phillipa}@icis.qut.edu.au

Abstract

In this paper the application of defeasible logic for automated negotiation is investigated. Defeasible logic is flexible enough to be adapted to several possible negotiation strategies, has efficient implementations, and provides a formal basis for analysis (e.g. to explain why a negotiation was not successful). Two case studies, one small and one more comprehensive, will be described and the feasibility of approaches based on defeasible logic will be discussed.

1 Introduction

It is well known that descriptions of real life scenarios are, very often, partial, and somewhat unreliable. However, we want to reason and draw conclusions about them. Classical logic, alas, is not very well suited to deal with such cases, because it needs complete, consistent, and reliable information; otherwise it could produce wrong and counterintuitive results.

To obviate the problem alluded to above, a plethora of non-monotonic systems, with different intuitions, have been put forward. Unfortunately, most of the proposed non-monotonic logics are computationally intractable (cf. [9]), and have been used only for a few standard examples, while real life applications require low complexity and more complicated cases (cf. [23]).

In this paper we discuss how to apply a particular non monotonic system (Defeasible Logic) to two case studies. First of all Defeasible Logic has been developed by Nute [24, 25] over several years with a particular concern about computational efficiency (indeed, its efficiency is linear cf. [20]) and ease of implementation (nowadays several implementations exist [10, 21] and some of them can deal with theories consisting of over 100,000 propositional rules [21]).

It is worth noting that Defeasible Logic has been successfully applied to the design and implementation of various kinds of controllers (cf. [10]); some scholars have argued in favor of its applicability in the context of normative

*This work was supported by an ARC SPIRT Grant "Self-describing transactions operating in a large, open, heterogeneous and distributed environment" between QUT and GBST Pty Ltd.

reasoning [26, 27, 1]. The case studies at hand in this paper are taken from a different field: negotiation and e-commerce.

In an e-commerce setting support for negotiation needs to be efficient, transparent, and expressive. It should be possible to specify negotiation strategies, tactics, and rules fairly straightforwardly. In the last few years the relevance of argumentation theory to capture negotiation and its protocols has been argued (cf., for example [29, 19]). However, in [14, 15], a close connection between argumentation theory and Defeasible Logic was established, and in [2, 3] it was shown that Defeasible Logic is flexible enough to be adapted to several possible argumentation strategies.

In Section 2 we shortly rehearse the basic of Defeasible Logic, and in Section 3 we discuss two case studies from e-commerce. In the first case study (Section 3.1) we examine how to use Defeasible Logic to describe brokered trades, and in the second case study (Section 3.2) we investigate bargaining. Finally in Section 4 we discuss shortly relted and future work.

2 Basics of Defeasible Logic

We begin by presenting the basic ingredients of defeasible logic (cf. [6]). A defeasible theory contains six different kinds of knowledge: facts, strict rules, defeasible rules, defeaters, a superiority relation, and a specification of complementary literals. We only consider rules that are essentially propositional. Rules containing free variables are interpreted as the set of their ground instances.

Facts are indisputable statements, for example, "Tweety is an emu". Written formally, this would be expressed as $emu(tweety)$.

Strict rules are rules in the classical sense: whenever the premises are indisputable (e.g. facts) then so is the conclusion. An example of a strict rule is "Emus are birds". Written formally:

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

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is "Birds typically fly"; written formally:

$$bird(X) \Rightarrow flies(X)$$

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.*

Defeaters are rules that cannot be used to draw any conclusions. Their only use is to prevent some conclusions. In other words, they are used to defeat some defeasible rules by producing evidence to the contrary. An example is "If an animal is heavy then it might not be able to fly". Formally:

$$heavy(X) \leadsto \neg flies(X)$$

The main point is that the information that an animal is heavy is not sufficient evidence to conclude that it doesn't fly. It is only evidence against the conclusion that a heavy animal flies. In other words, we don't wish to conclude $\neg flies$ if $heavy$, we simply want to prevent a conclusion $flies$.

The *superiority relation* among rules is used to define priorities among rules, that is, where one rule may override the conclusion of another rule. For example, given the facts

$$\begin{array}{l} \rightarrow bird \\ \rightarrow brokenWing \end{array}$$

and the defeasible rules

$$\begin{array}{rrl} r: & bird & \Rightarrow flies \\ r': & brokenWing & \Rightarrow \neg flies \end{array}$$

which contradict one another, no conclusive decision can be made about whether a bird with a broken wing can fly. But if we introduce a superiority relation $\succ$ with $r' \succ r$, then we can indeed conclude that the bird cannot fly. The superiority relation is required to be acyclic.

For each literal p we define the set of *p-Complementary literals* ($\mathcal{C}(p)$), that is, the set of literals that cannot hold when p does. Let us consider an example; let us suppose we have the predicates *married* and *bachelor*. Here, we define, for any constant a, $\mathcal{C}(married(a)) = \{\neg married(a), bachelor(a)\}$. We know that, under the usual interpretation of the predicates they cannot be true at the same time for one and the same individual. We stipulate that the negation of a literal is always complementary to the literal.

Now we present formally defeasible logics. A *rule* r consists of its *antecedents* (or *body*) $A(r)$ which is a finite set of literals, an arrow, and its *consequent* (or *head*) $C(r)$ which is a literal. There are three kinds of arrows, $\rightarrow$, $\Rightarrow$ and $\leadsto$ which correspond, respectively, to strict rules, defeasible rules and defeaters. Where the body of a rule is empty or consists of one formula only, set notation may be omitted in examples.

Given a set R of rules, we denote the set of all strict rules in R by R_s, the set of strict and defeasible rules in R by R_{sd}, the set of defeasible rules in R by R_d, and the set of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q, and $R[\mathcal{C}(q)]$ denotes the set of rules in R whose consequent is in $\mathcal{C}(q)$.

A *defeasible theory* D is a structure

$$D = (F, R, \succ, \mathcal{C})$$

where F is a finite set of facts, R is a finite set of rules, $\succ$ is a binary relation over R, and $\mathcal{C}$ is a function mapping a literal to a set of literals.

A *conclusion* of D is a tagged literal, where a tag is either ∂ or Δ, that may have positive or negative polarity.

$+\Delta q$ which is intended to mean that q is definitely provable in D (i.e., using only strict rules).

$-\Delta q$ which is intended to mean that we have proved that q is not definitely provable in D.

$+\partial q$ which is intended to mean that q is defeasibly provable in D.

$-\partial q$ which is intended to mean that we have proved that q is not defeasibly provable in D.

Provability is based on the concept of a *derivation* (or *proof*) in $D = R$. A derivation is a finite sequence $P = (P(1), \dots P(n))$ of tagged literals satisfying four conditions (which correspond to inference rules for each of the four kinds of conclusion). In the following $P(1..i)$ denotes the initial part of the sequence P of length i.

$+\Delta$:
If $P(i+1) = +\Delta q$ then
$\quad \exists r \in R_s[q]$
$\quad\quad \forall a \in A(r) : +\Delta a \in P(1..i)$

$-\Delta$:
If $P(i+1) = -\Delta q$ then
$\quad \forall r \in R_s[q]$
$\quad\quad \exists a \in A(r) : -\Delta a \in P(1..i)$

The definition of Δ describes just forward chaining of strict rules. For a literal q to be definitely provable we need to find a strict rule with head q, of which all antecedents have been definitely proved previously. And to establish that q cannot be proven definitely we must establish that for every strict rule with head q there is at least one antecedent which has been shown to be non-provable.

Now we turn to the more complex case of defeasible provability.

$+\partial$: If $P(i+1) = +\partial q$ then either
(1) $+\Delta q \in P(1..i)$ or
(2) (2.1) $\exists r \in R_{sd}[q] \forall a \in A(r) : +\partial a \in P(1..i)$ and
(2.2) $\forall p \in \mathcal{C}(q) - \Delta p \in P(1..i)$ and
(2.3) $\forall s \in R[\mathcal{C}(q)]$ either
(2.3.1) $\exists a \in A(s) : -\partial a \in P(1..i)$ or
(2.3.2) $\exists t \in R_{sd}[q]$ such that
$\forall a \in A(t) : +\partial a \in P(1..i)$ and $t > s$

$-\partial$: If $P(i+1) = -\partial q$ then
(1) $-\Delta q \in P(1..i)$ and
(2) (2.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r) : -\partial a \in P(1..i)$ or
(2.2) $\exists p \in \mathcal{C}(q)$ such that $+\Delta p \in P(1..i)$ or
(2.3) $\exists s \in R[\mathcal{C}(q)]$ such that
(2.3.1) $\forall a \in A(s) : +\partial a \in P(1..i)$ and
(2.3.2) $\forall t \in R_{sd}[q]$ either
$\exists a \in A(t) : -\partial a \in P(1..i)$ or $t \not> s$

Let us work through the condition for $+\partial$, an analogous explanation holds for $-\partial$. To show that q is provable defeasibly we have two choices: (1) We show that q is already definitely provable; or (2) we need to argue using the defeasible part of D as well. In particular, we require that there must be a strict or defeasible rule with head q which can be applied (2.1). But now we need to consider possible "attacks", that is, reasoning chains in support of a complementary of q. To be more specific: to prove q defeasibly we must show that every complementary literal is not definitely provable (2.2). Also (2.3) we must consider the set of all rules which are not known to be inapplicable and which have head in $\mathcal{C}(q)$ (note that here we consider defeaters, too, whereas they could not be used to support the conclusion q; this is in line with the motivation of defeaters given in subsection 2.1). Essentially each such rule s attacks the conclusion q. For q to be provable, each such rule s must be counterattacked by a rule t with head q with the following properties: (i) t must be applicable at this point, and (ii) t must be stronger than s. Thus each attack on the conclusion q must be counterattacked by a stronger rule.

3 Negotiation and Defeasible Logic: Two Case Studies

In this section we present in details two examples of the application of Defeasible Logic to automated negotiation scenarios. In the first example (Section 3.1) we consider a case of brokered trade, and we show how to use Defeasible Logic (1) to select goods against a set of constraints, and (2) to choose the most appropriate good. Then (Section 3.2) we consider a simple case of negotiation: single issue bargaining. Here Defeasible Logic is used both in the protocol and in the dispute phases.

3.1 Case Study 1: Brokered Trade

Brokered trades take place via an independent third party (broker). The broker matches both buyers' and sellers' requirements, and will propose a transaction when all parties can be satisfied by the trade.

Suppose we have the following scenario: Andrew contacts a broker, he wants to buy a yacht under 5 years old, at least 60 ft for around $130,000. He is flexible and will pay more (with a limit of $150,000) for a younger or bigger yacht. He thinks that each extra foot is worth $1000 and each year of age less than 5 years adds $3000 to the value of the yacht.

He rates the importance of achieving the price level of $130,000 more than the importance of either the age or size of the yacht. However, he prefers a younger boat over a longer one.

The broker has five yachts to sell: the first is 60 ft and 2 years old, the price is $140,000. The second is 70 ft and 1 year old, the price is $150,000. The third yacht is 100 ft and 5 years old with a price of $170,000. The fourth is 3 years old and 50 ft with a price of $125,000. Finally the fifth is 80 ft and 3 years old with a price of $150,000.

The above data can be summarized in Table 1. The scenario presented

Yacht	Length	Age	Price (000)
1^{st}	60	2	140
2^{nd}	70	1	150
3^{rd}	100	5	170
4^{th}	50	3	125
5^{th}	80	3	150

Table 1: Yacht details

above can be represented as a two stage process. In the first phase we filter the yachts according to their characteristics and against Andrew's desiderata. At this point the most appropriate boat can be chosen. This process can be represented formally in terms of two correlated defeasible theories: the first for filtering and the second for choosing.

The language of the first defeasible theory ($D_f = (F_f, R_f, \succ_f, \mathcal{C}_f)$) includes the following terms and predicates:

- *length*(x) the length of the yacht;

- $age(x)$ how old the yacht is;
- $offer(x, y)$ meaning how much (y) Andrew is ready to pay for the yacht x;
- $price(x, y)$ meaning the price (y) of the yacht x;
- $buyable(x)$ meaning whether the yacht x meets Andrew's conditions;

The set of facts (F_f) is just the set of predicates that can be deduced from Table 1, while the rules (R_f) can be expressed as follows:

1. $f_1 : length(x) < 60ft \Rightarrow \neg buyable(x)$
2. $f_2 : age(x) > 5yrs \Rightarrow \neg buyable(x)$
3. $f_3 : price(x, y), y > \$150 \Rightarrow \neg buyable(x)$
4. $f_4 : \Rightarrow offer(x, \$130)$
5. $f_5 : length(x) > 60ft \Rightarrow offer(x, (\$130 + g(length(x))))$
6. $f_6 : age(x) < 5yrs \Rightarrow offer(x, (\$130 + h(age(x))))$
7. $f_7 : age(x) < 5yrs, length(x) > 60ft \Rightarrow offer(x, (\$130 + g(length(x)) + h(age(x))))$
8. $f_8 : offer(x, y) \Rightarrow buyable(x)$
9. $f_9 : offer(x, y), price(x, z), y < z \Rightarrow \neg buyable(x)$

where the superiority relation $\succ_f$ is thus defined: $f_1 \succ f_8$, $f_2 \succ f_8$, $f_3 \succ f_8$, $f_9 \succ f_8$ as far as the rules for *buyable* are concerned, and $f_5 \succ f_4$, $f_6 \succ f_4$, $f_7 \succ f_4$, $f_7 \succ f_5$, and $f_7 \succ f_6$ for *offer*.

Finally $\mathcal{C}$ maps each literal to its negation. Moreover for each x, y we have that $offer(x, z)$, such that $z \neq y$ is in $\mathcal{C}(offer(x, y))$.

The first three rules state the minimal requirements; the fourth rule sets the basic offer for a generic boat, while f_5, f_6 and f_7 represent Andrew's "flexibility" refining the offer for a specific bigger or younger yacht. Rule f_8 says two things 1) that the yacht x meets Andrew's requirements, and that Andrew has set an offer of y for that yacht. However, rule f_9 is used to see whether the offer matches the price of the yacht.

The condition on $\mathcal{C}$ states that there is at most one unique offer for a given yacht.

The second theory (D_c), for choosing, requires the following additional predicates:

- $min\_price(x)$ meaning that the yacht x is the cheapest of the selected ones;
- $min\_age(x)$ saying that the yacht x is the youngest of the selected ones;
- $max\_length(x)$ which is true for the longest yacht;
- $buy(x)$ saying that x is the most suitable candidate.

The rules for selecting the most appropriate boat are:

- $c_1 : min\_price(x) \Rightarrow buy(x)$

- $c_2 : min\_age(x) \Rightarrow buy(x)$
- $c_3 : max\_length(x) \Rightarrow buy(x)$

where the superiority relation is $c_1 \succ c_2$, $c_1 \succ c_3$, and $c_2 \succ c_3$. Since only one boat has to be selected the complementary literals area as follows: $\mathcal{C}(buy)(x) = \{\neg buy(x)\} \cup \{buy(y) | y \neq x\}$.

We are now ready to present the filtering and choosing processes in detail. Let us consider the first yacht. The first three rules are not applicable for it, thus to see whether it is a suitable boat we have to calculate its value. Rules f_5 and f_7 are not applicable. On the other hand, both f_4 and f_6 are applicable, but f_6 is the strongest of the two, and no applicable rule defeats it, therefore we can conclude $+\partial offer(1, \$139)$. At this point f_9 becomes applicable, it defeats f_8, but it is not defeated by it, thus we can derive $-\partial buyable(1)$. So, we have that the first yacht is too expensive and it is not a suitable candidate for the deal.

It is immediate to see that we can derive $-\partial buyable(3)$, and $-\partial buyable(4)$. The former because rule f_1 is applicable, that is the boat is too short, and the latter because rule f_3 is applicable, that is, its price is out of the price range.

The first three rules are not applicable for 2 and 5. Thus, similarly to what we have done in the previous case, we have to determine their values, and we have, using r_7, $+\partial offer(2, \$152)$ and $+\partial offer(5, \$156)$, which makes r_9 applicable for both yachts. Then $+\partial buyable(2)$ and $+\partial buyable(5)$. Thus the second and the fifth yachts are ones that will be used in the second phase.

The actual defeasible theory we obtain in this case consists of the following instances of rules

- $r_1 : min\_price(2) \Rightarrow buy(2)$
- $r_2 : min\_price(5) \Rightarrow buy(5)$
- $r_3 : min\_age(2) \Rightarrow buy(2)$
- $r_4 : min\_age(5) \Rightarrow buy(5)$
- $r_5 : max\_length(2) \Rightarrow buy(2)$
- $r_6 : max\_length(5) \Rightarrow buy(5)$

where $r_{1,2} \succ r_{3,4,5,6}$, $r_{3,4} \succ r_{5,6}$.

The complementary literals are thus defined: $\mathcal{C}(buy(2)) = \{\neg buy(2), buy(5)\}$, and $\mathcal{C}(buy(5)) = \{\neg buy(5), buy(2)\}$, meaning that only one yacht will be bought.

The first two rules are not applicable since there is not a boat with the lowest price; rules r_4 and r_5 are not applicable since 5 is not the youngest yacht and 2 is not the longest. Then the only applicable rules are r_3 and r_6, but we know that r_3 is stronger than r_6. So we derive $+\partial buy(2)$. This means that the second yacht is the most appropriate one according to Andrew's conditions.

3.2 Case Study 2: Simple Negotiation (Bargaining)

A negotiation is a discussion between parties for the purpose of reaching an agreement. This suggests representing a negotiation as a dialogue between parties, this dialogue is articulated in progressive stages, where the parties make offers, reject or accept offers, or propose counter-offers. In this view it can be thought of as a special kind of argumentation where we have two different aspects: the protocol and the content of the negotiation (cf. [29]). The protocol describes the rules of the dispute, for example how the parties exchange their offers, and how and when the negotiation can go on or terminate.

Here, for the sake of simplicity, we consider only one-to-one negotiations, that is, negotiations where only two parties are involved —let us call them the Proponent and the Opponent. In this perspective we can formally represent a negotiation as three sequences of defeasible theories. The first sequence records the evolution of the protocol, while the second and the third theories are used to store the knowledge bases or defeasible theories (DT) of the two parties.

Graphically a negotiation can be depicted as follows

Stage 1	Protocol$_1$	Proponent DT$_1$	Opponent DT$_1$
Stage 2	Protocol$_2$	Proponent DT$_2$	Opponent DT$_2$
⋮	⋮	⋮	⋮
Stage n	Protocol$_n$	Proponent DT$_n$	Opponent DT$_n$
⋮	⋮	⋮	⋮

AGREEMENT / END OF NEGOTIATION

Another element we have to take into account is the negotiation strategy, that is the mechanism for passing from one stage to the next. Several negotiation strategies can be devised, for example:

- single fixed theory: a party uses a single defeasible theory through the whole negotiation, which is evaluated using new data that becomes available during the negotiation.
- fixed sequence of theories; here a party fixes a sequence of theories for the whole negotiation.
- parameterized theories: a party defines a set of rules that can be triggered or modified according to the stage of the negotiation.
- revision of theories: a party modifies the actual theory from stage to stage according to the result of the previous stage.

Finally we have to specify how the offers are exchanged between the parties. First of all the parties do not have to disclose every piece of information they have, thus we partition the defeasible theory of a party into two parts: the public part, whose conclusions have to be disclosed to the other party, and a private part. The Proponent computes its theory obtaining a set of conclusions and the public conclusions are passed to the Opponent that uses them to supplement its facts; at this point the Opponent theory is computed. According to the result of this last computation we can have three possible results: the Opponent accepts the Proponents offer and the negotiation is terminated successfully; the Opponent rejects the Proponents offer, makes a counter offer and the negotiation is continued (i.e., we pass to the next stage); or the Opponent rejects the offer, but the two parties cannot converge on an agreement so there is no point in negotiating, and the negotiation is terminated with a failure.

We illustrate with a simple bargaining example.

Mary is interested in buying a computer system advertised for \$1300. She wants to negotiate to buy the advertised system at a lower price. The seller has a cash-flow crisis and is keen to make sales immediately. They estimate the loss of a sale to a customer in the shop costs them \$50 in bank interest and advertising costs. The total cost of the system components is \$1000. We assume for simplicity that the features of the system are fixed, and the negotiation is conducted over a single issue; the price of the system.

Tactics are the practical expression of strategies and we represent them as functions that generate new offers. These functions can be simple or complex depending on the strategy and the amount of information they incorporate. Complex tactics can include the negotiation rules and the history of previous offers. We represent tactics with the $f(x, y)$ predicate. Possible tactic functions available to the seller in this scenario include:

- reducing the previous offer by a fixed amount or fixed percentage
- reducing the previous offer by the same amount or percentage evidenced by the buyers offers
- reducing the margin over cost
- increasing the discount
- changing the configuration
- changing the terms of sale

Many of these tactics can also be used by the buyer.

The predicates for this simple negotiation are:

- $cost(x, y)$ the cost y of the system x.
- $sellerPrice(x, y)$ the price y of the system x.
- $buyerOffer(x, y)$ how much y the buyer is willing to pay for the system x.
- $minimumPrice(x, y)$ the sellers minimum price.
- $maximumPrice(x, y)$ the buyers maximum price.
- *immediate* whether or not payment is made immediately or delayed.
- $f(x, y)$ a tactic function used to generate the next offer price y for the system x.

Protocol Rules

- $p_1 : \Rightarrow negotiate$
- $p_2 : negotiate, step(n) \Rightarrow step(n + 1)$
- $p_3 : buyerOffer^t(x, y), sellerPrice^t(x, z), |y - z| > \$500 \Rightarrow \neg negotiate$
- $p_4 : acceptable \Rightarrow \neg negotiate$

Rule s_6 and rule b_5 show that offers are ordered over time and the next prices for buyers and sellers are based on their previous price and the selected tactic function. In general, we would expect the sellers tactic function to reduce, and the buyers tactic function to increase the price offered in each round. In order to stop fruitless negotiation p_3 allows for negotiations to cease if the offers diverge by an arbitrary amount, \$500 in this case. Rule p_4 ends the negotiations when an offer is accepted.

Rules p_1 and p_2 guide the negotiation process, if the negotiation is not terminated due to failure or completion, then proceed to the next step.

The tactic functions themselves may also change during the negotiation process. Changes may be in response to what is learned about the opponents preferences and tactics, the time remaining to negotiate, or new information received during the negotiation.

Seller Rules

- $s_1 : \mathit{cost}(x, y) \Rightarrow \mathit{sellerPrice}^{t=0}(x, y + \$300)$
- $s_2 : \mathit{cost}(x, y) \Rightarrow \mathit{minimumPrice}(x, y + \$20)$
- $s_3 : \mathit{cost}(x, y), \mathit{immediate} \Rightarrow \mathit{minimumPrice}(x, y - \$50)$
- $s_4 : \mathit{buyerOffer}^{t-1}(x, y), \mathit{sellerPrice}^{t}(x, z), y \geq z \Rightarrow \mathit{acceptable}$
- $s_5 : \mathit{buyerOffer}^{t-1}(x, y), \mathit{minimumPrice}(x, z), y < z \Rightarrow \neg \mathit{acceptable}$
- $s_6 : \mathit{sellerPrice}^{t-1}(x, y) \Rightarrow \mathit{sellerPrice}^{t}(x, f(x, y))$

Rule s_1 provides the basis for the sellers initial price and rules s_2 and s_3 define the sellers minimum price, in terms of the cost of the system. Rule s_3 also provides an additional discount for an immediate payment.

The two rules s_4 and s_5 describe the sellers strategy for continuing or terminating the negotiation. Rule s_4 describes an ideal situation where the buyer offers to pay more than the next price the seller is willing to offer, the buyers offer can be accepted and the negotiation concluded successfully. If the buyers offer is less than the sellers next offer and rule p_5 does not apply, the negotiation will continue with the seller offering $\mathit{sellerPrice}^{t}(x, z)$.
Rule s_5 defines an unacceptable offer as one that is below the minimum price. In this case the seller can either make their next (higher) counter-offer or terminate the negotiation according to rule p_3.

The superiority relation $\succ$ is defined for the seller as: $s_3 \succ s_2$. This relation shows the seller will accept an offer below their cost, if the buyer is willing to pay immediately, however $s_5 \succ s_4$ ensures the seller will not sell at less than the minimum price.

Buyer Rules

- $b_1 : \mathit{sellerPrice}^{t=0}(x, y) \Rightarrow \mathit{maximumPrice}(x, y - \$200)$
- $b_2 : \mathit{sellerPrice}^{t=0}(x, y) \Rightarrow \mathit{buyerOffer}^{t=0}(x, y - \$400)$
- $b_3 : \mathit{sellerPrice}^{t}(x, y), \mathit{buyerOffer}^{t}(x, z), y \leq z \Rightarrow \mathit{acceptable}$
- $b_4 : \mathit{sellerPrice}(x, y), \mathit{maximumPrice}(x, z), y > z \Rightarrow \neg \mathit{acceptable}$

- $b_6 : \mathit{buyerOffer}^{t-1}(x, y) \Rightarrow \mathit{buyerOffer}^{t}(x, f(x, y))$

The buyers maximum price is based on the sellers advertised price and is described by rule b_1. The arbitrary reduction of $200 provides a basis for the negotiation. Similarly the buyers initial offer (b_2) includes an opening ambit claim of $400 as a counter-offer to the sellers opening price. Rules b_3 and b_4 describe continuation rules similar to the sellers rules s_4 and s_5.

The superiority relation $\succ$ is defined for the Buyer as: $b_4 \succ b_3$. The relation shows the Buyer will not accept an offer over its maximum price.

The negotiation process is guided by rules p_1 and p_2, and is started by the buyer making their offer $\mathit{buyerOffer}^{t=0}$. The seller evaluates p_1, p_2, and then evaluates the offer according to rules s_4 and s_5. The seller can either accept the offer or make the counter offer $\mathit{sellerPrice}^{t}(x, f(x, y))$. The buyer proceeds in the same fashion evaluating p_1 and p_2, and offers against rules b_3 and b_4. The exchange of offers continues until either: s_4 or b_3 are true, and the negotiation is successfully concluded. Or p_3, s_5 or b_4 cause the negotiation to terminate without a result.

The negotiation starts at stage 0 ($t = 0$), and each stage consists of two steps: the Proponent (Seller) move and the Opponent (Buyer) move. Here the Seller discloses the initial price ($\mathit{sellerPrice}^{t=0}$), calculated from s_1 using the cost of the system (a private piece of information of the Seller); at the same time the private literals *minimumPrice* is derived. At this point the protocol becomes active to determine whether the negotiation can continue. The only applicable rule is p_{xx}, from which $+\partial \mathit{negotiate}$ can be derived; thus the Opponent move is triggered. The Buyer adds the public data of the Seller to its own facts and computes the resulting theory, where the public data $\mathit{buyerOffer}^{t=0}$ as well as the private conclusion *maximumPrice* are derived.

The protocol is fed with the new Buyer public data to decide whether the negotiation should continue. The Seller price and the Buyer offer are compared, and if they are not too distant, according to a predefined parameter, the negotiation moves to the next stage (rules p_3 and p_2).

Again the first step of the new stage is up to the Proponent, which adds public conclusion of the Opponent to its theory. If the Buyer offer ($\mathit{buyerOffer}^{t=0}$), a new fact, is better than the current proposal ($\mathit{sellerPrice}^{t=1}$ calculated from s_6 according to the tactic function f and the previous offer $\mathit{sellerPrice}^{t=0}$), then it accepts the offer and the protocol terminates the negotiation successfully. Otherwise a new proposal is made (rule s_6), unless the limit is reached (rule s_5).

A similar procedure is actuated by the Buyer, in case of a new offer from the Seller.

This simple scenario shows how defeasible logic can be used to describe both protocols and strategies. More complex two party multi-issue negotiations can be described by additional rules. These rules could provide values for each issue, the relative importance of issues to one another and the preferred trade-offs between issues.

4 Related and Future Work

There are several other logical approaches to two party negotiation [33, 22, 28]. The work by Tohme [33] suggests defeasible reasoning can be used to evaluate and generate offers, however this paper concentrates on defining the negotiation process rather than the use of defeasible reasoning for evaluation. Matos and Sierra [22] suggest Case-Based or Fuzzy reasoning approaches. The use of Fuzzy rules is a similar approach to ours, but does not allow rules to be overridden in exceptional cases as does defeasible reasoning. Parsons et al. [28] use logical arguments (argumentation) to support or "undercut" offers rather than for the evaluation or generation of offers.

Grosof et al. [16] use Courteous Logic Programs (CLP) to define and prioritize business rules. This work is extended in Reeves et al. [31] with CLP's used to express knowledge about user preferences, constraints, and negotiation structures (auctions in this case). Antoniou et al. [4] have shown that defeasible theories are at least as expressive as CLP's.

Faratin, Jennings and Sierra et al [32, 11, 12, 17] use *Value functions*, a version of Multi-Attribute Utility Theory (MAUT) [30] to generate and evaluate offers in multi-attribute negotiations. While value functions are effective for analyzing the acceptability of offers, we know people have problems with identifying and defining utility functions [18, 8]. In this respect defeasible reasoning provides a more natural way of expressing preferences and goals, and the possible trade-offs between them.

Barbuceanu [5] extends the MAUT approach with *acceptability constraints* and constraint optimization. Acceptability constraints are used to define acceptable combinations of attributes and attribute values. We believe this approach using both MAUT principles and defeasible reasoning provides an excellent way for us to extend our work to encompass multi-attribute negotiation.

However, as it has been pointed out in [1], Defeasible Logic alone is not able to deal with real-life cases, but it has to be supplemented with other formalisms and tools, such as: arithmetical capabilities, temporal logic, etc. One solution to the problem we have just alluded to is to use labels [13, 34] to encode such additional features. We can have two levels of labels: labelling the rules, for example with event-label; or we can use labels inside the language, that is the literals occurring in a defeasible theory are just propositional wrappers for complex entities. The first type of labels will enable a better and more general treatment of the evolution of a negotiation and the steps it involves.

The use of labels and more negotiation strategies, for example using revision procedures for the defeasible theories using the methodology of [7], have to be devised. These and other topics connecting automated negotiation and Defeasible Logic will be the subject of future works.

Acknowledgments

We would like to thank Grigoris Antoniou, David Billington, Marlon Dumas Antonino Rotolo for fruitful discussions on defeasible logic and its application to e-commerce, automated negotiation and other fields.

References

[1] Grigoris Antoniou, David Billington, Guido Governatori, and Michael Maher. On the modeling and analysis of regulations. In *Proceedings of the Australian Conference on Information Systems*, 1999.

[2] Grigoris Antoniou, David Billington, Guido Governatori, and Michael Maher. A flexible framework for defeasible logics. In *Proceedings of the American National Conference on Artificial Intelligence (AAAI-2000)*, pages 405–410. AAAI Press, 2000.

[3] Grigoris Antoniou, David Billington, Guido Governatori, Michael Maher, and Andrew Rock. A family of defeasible reasoning logics and its implementation. In Werner Horn, editor, *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, Amsterdam, 2000. IOS Press.

[4] Grigoris Antoniou, Michael J. Maher, and David Billington. Defeasible logic versus logic programming without negation as failure. *Journal of Logic Programming*, 41:45–57, 2000.

[5] Mihai Barbuceanu and Wai-Kau Lo. A multi-attribute utility theoretic negotiation architecture for electronic commerce. In *Proceedings Agents 2000*, pages 239–246, Barcelona, Spain, 2000. ACM, ACM Press.

[6] David Billington. Defeasible logic is stable. *Journal of Logic and Computation*, 3:370–400, 1993.

[7] David Billington, Grigoris Antonoiu, Guido Governatori, and Michael J. Maher. Revising nonmonotonic belief sets: The case of defeasible logic. In W. Burgard, T. Christaller, and A. Cremers, editors, *KI-99: Advances in Artificial Intelligence*, volume 1701 of *LNAI*, Berlin, 1999. Springer Verlag.

[8] Katrin Borcherding, Thomas Epple, and Detlof Von Winterfeldt. Comparison of weighting judgments in multi-attribute utility measurement. *Management Science*, 37(12):1603–1619, December 1991.

[9] Marco Cadoli and Marco Schaerf. A survey of complexity results for nonmonotonic logics. *Journal of Logic Programming*, 17:127–160, 1993.

[10] Michael Covington, Donald Nute, and A. Vellino. *Prolog Programming in Depth*. Prentice Hall, 1997.

[11] Peyman Faratin, Carles Sierra, and Nick R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24(3–4):159–182, 1998.

[12] Peyman Faratin, Carlos Sierra, Nick R. Jennings, and P. Buckle. Designing responsive and deliberative automated negotiators. In *Proceedings AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, pages 12–18, Orlando, Florida, 1999. AAAI, AAAI Press.

[13] Dov M. Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996.

[14] Guido Governatori and Michael J. Maher. An argumentation-theoretic characterization of defeasible logic. In Werner Horn, editor, *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, Amsterdam, 2000. IOS Press.

[15] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logics. In Riichiro Mizoguchi and John Slaney, editors, *PRICAI 2000: Topics in Artificial Intelligence*, volume 1886 of *LNAI*, pages 27–37, Berlin, 2000. Springer Verlag.

[16] Benjamin N. Grosof, Yannis Labrou, and Hoi Y. Chan. A declarative approach to business rules in contracts: Courteous logic programs in XML. In *Proceedings of the 1st ACM Conference on Electronic Commerce (EC99)*, Denver, Colorado, November 1999. ACM, ACM Press.

[17] Nick R. Jennings, Simon Parsons, Carlos Sierra, and Peyman Faratin. Automated negotiation. In *Proceedings of the 5th International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Systems, PAAM*, pages 23–30, 2000.

[18] Gregory E. Kersten and Stan Szpakowicz. Modelling business negotiations for electronic commerce. In *Intelligent Information Systems: Proceedings of the 7th Workshop*, pages 17–28, Warsaw, 1998. IPI PAN.

[19] Ronald Loui. Process and policy: resource-bounded non-demonstrative reasoning. *Computational Intelligence*, 14:1–38, 1998.

[20] Michael J. Maher. Propositional defeasible logic has linear complexity. Technical report, Department of Mathematical and Computer Sciences, Loyola University, Chicago, 2000.

[21] Michael J. Maher, Andrew Rock, Grigoris Antoniou, David Billignton, and Timothy Miller. Efficient defeasible reasoning systems. In *Proc. Interantional Conference on Tool in Artificial Intelligence*, 2000.

[22] Noyda Matos and Carles Sierra. Evolutionary computing and negotiating agents. In *Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98)*, pages 91–111, St Paul, Minneapolis, May 9–13 1998.

[23] Leora Morgensten. Inheritance comes of age: Applying nonmonotonic techniques to problems in industry. *Artificial Intelligence*, 103:1–34, 1998.

[24] Donald Nute. Defeasible reasoning. In *Proc. 20^{th} Hawaii International Conference on System Science*, pages 470–477. IEEE Press, 1987.

[25] Donald Nute. *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, chapter Defeasible Logic, pages 353–395. Oxford University Press, Oxford, 1994.

[26] Donald Nute. Apparent obligation. In Donald Nute, editor, *Defeasible Deontic Logic*, pages 287–316. Kluwer, 1997.

[27] Donald Nute. Norms, priorities and defeasibility. In Paul McNamara and Henry Prakken, editors, *Norms, Logics and Information Systems. New Studies in Deontic Logic*, pages 83–100. IOS Press, Amsterdam, 1998.

[28] Simon D. Parsons, Carlos A. Sierra, and Nick R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[29] Henry Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, Forthcoming.

[30] Howard Raiffa. *The Art and Science of Negotiation.* Harvard University Press, Cambridge, MA, 1982.

[31] Daniel M. Reeves, Michael P. Wellman, Benjamin N. Grosof, and Hoi Y. Chan. Automated negotiation from declarative contract descriptions. In *Seventeenth National Conference on Artificial Intelligence, Workshop on Knowledge-Based Electronic Markets(KBEM)*, Austin, Texas, July 30–31 2000. AAAI, AAAI Press.

[32] Carles Sierra, Peyman Faratin, and Nick R. Jennings. A service-oriented negotiation model between autonomous agents. In *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'97)*, pages 17–35, Ronneby, Sweden, 1997.

[33] Fernando Tohme. Negotiation and defeasible reasons for choice, March 24-26 1997. Presented at the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, Stanford University, California.

[34] Doug Walton. Applying labelled deductive systems and multi-agent systems to source-based argumentation. *Journal of Logic and Computation*, 9:63–80, 1998.