Informal introduction to defeasible logic

A defeasible theory (a knowledge base in Defeasible logic) consists of five different type of knowledge bases: facts, strict rules, defeasible rules, defeaters, and a superiority relation.

Facts are indisputable statements, for example “Tweets in an emu”. Formally written as *emu(tweets)*

Strict rules are the rules from the good old classical logic i.e. whenever the premises are indisputable (e.g., facts) then so is the conclusion.

$$emu(x) \rightarrow bird(x)$$

Defeasible rules are the rules that can be defeated by the contrary evidence. An evidence of such a rule is “Birds typically fly” written formally as,

$$Bird(x) \Rightarrow flies(X)$$

The main purpose is that if we know something is bird, then we may conclude that it flies **unless** there is other, superlative or inferior argument suggesting it cannot fly.

Defeaters are rules that cannot be used to draw any conclusions. Their only use is to prevent some conclusions. Moreover, they are used to defeat some defeasible rules by producing contrary evidence to it. An example would be “If an animal is heavy then it might not be able to fly”. Formally written as

$$Heavy(X) \rightsquigarrow \neg flies(X)$$

It is the only evidence that animal may not be able to fly. In other words, we do not wish to conclude *¬flies(X)* if *heavy(X);* we simply want to prevent the conclusion of *flies(X).*

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

$$r: \quad bird(X) \Rightarrow flies(X)$$

$$r': \quad brokenwing(X) \Rightarrow \neg flies(X)$$

which contradict with each other, no conclusive decision can be made about whether a bird with broken wings can fly. But if we introduce a superiority relation > with *r’ > r,* with the intended meaning that the r’ is strictly stronger than r, then we can indeed conclude that bird cannot fly.

Note that the cycle in the superiority relation is counterintuitive. In the above example, it makes no sense to have both *r > r’ and r’ >r.* Consequently, we will focus on cases where the superiority relation is acyclic. Another point worth noting is that, in Defeasible Logic, priorities are local in the following sense: two rules are competing with one another only if they have complementary heads.

Thus, since the superiority relation is used to resolve conflicts among competing rules, it is only used to compare rules with complementary heads; the information r > r’ for rules r, r’ without complementary heads may be part of the superiority relation, but has no effect on the proof theory.

Why the need of nonmonotonic reasoning or defeasible logic?

Because monotonic reasoning is too restrictive. Monotonic reasoning in general and truth preserving reasoning in particular work primarily to prevent us from reaching false conclusions. It only allows us to reach conclusions that we could not possibly doubt so long as our original reasons remain intact. It can be dangerous to believe things that are false, but it can be just as dangerous not to believe things that are true. We need a reasoning system that lets us draw likely conclusions with less than conclusive evidence.

Defeasible logic and its Language

We define atomic formulas in the usual way. A literal is any atomic formula or its negation. All and only literals are formulas of our language. Where ϕ is an atomic formula, we say ϕ and ∼ϕ are the complements of each other. ¬ϕ denotes the complement of any formula ϕ, positive or negative.

Rules are a class of expressions distinct from formulas. Rules are constructed using three primitive symbols: →, ⇒, and ⇝ .Where $A \cup \{\varphi\}$ is a set of formulas, $A \rightarrow \varphi$ is a strict rule, $A \Rightarrow \varphi$ is a defeasible rule, and A⇝ϕ is an undercutting defeater. In each case, we call A the antecedent of the rule and we call ϕ the consequent of the rule. Where $A = \{\psi\}$, we denote $A \rightarrow \varphi$ as $\psi \rightarrow \varphi$, and similarly for defeasible rules and defeaters. Antecedents for strict rules and defeaters must be non-empty; antecedents for defeasible rules may be empty.

Rules are defined over a language Σ, the set of propositions(atoms) and labels that may be used in the rule.

Definition 1.1. *A(r) ,→ C(r) consists of its unique label r, its antecedent A(r) (A(r) may be omitted if it is the empty set) which is a finite set of literals, an arrow ,→ (which is a placeholder for concrete arrows to be introduced in a moment), and its head (or consequent) C(r) which is a literal* .

The three kinds of rules, each represented in a different arrow as shown above. Given a set of 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 set of defeaters in R by $R_{dft.}$ R[q] denotes the set of rules with consequent R.

Definition 1.2. A superiority relation on R is a relation > on R. Where *r1 > r2, then r1 is superior to r2, and r2 is inferior to r1. Intuitively, r1 > r2 expresses that r1 overrules r2, should both rules be applicable.*

We assume > to be acyclic (that is, the transitive closure of > is irreflexive)

Definition 1.3. *A defeasible theory D is a triple (F, R,>) where F is a finite set of literals (called facts), R a finite set of rules, and* $> \subseteq R \times R$ *an acyclic superiority on R.*

Definition 1.4. (Regular form of Defeasible Theory). *A defeasible theory D = (F,R,>) is regular (or regular form) if the following three conditions are satisfied.*

1. Every literal is defined either solely by strict rules, or by one strict rule and other non-strict rules
2. No strict rule participates in the superiority relation >.
3. $F = \emptyset$.

Proof theory in defeasible logic

A *conclusion* in D (defeasible logic) is a tagged literal and can have one of the following four forms:

+Δq which is intended to mean that q is definitely provable in D.

-*Δq* which is intended to mean that we have proved that q is not definitely provable in D.

+∂q which is intended to mean that q is defeasibly provable in D.

-∂q which is intended to mean that we have proved that q is not defeasibly provable is D.

One important conclusion from the above consequence is that: if we are able to prove q definitely, then q is also defeasibly provable. This observation follows from the default logic.

Provability is defined on the concept of derivation (or proof) in *D = (F, R, >).* A derivation is a finite sequence *P = (P (1), ……. P(n))* of tagged literals satisfying the conditions (*P (1...i) denotes the initial part of the sequence of length i):*

$$
\begin{aligned}
&+\Delta: \text{If } P(i+1) = +\Delta q \text{ then either}\\
&\qquad q \in F \text{ or}\\
&\qquad \exists r \in R_s[q] \forall a \in A(r): +\Delta a \in P(1..i)
\end{aligned}
$$

The above derivation denotes that to prove *+Δq* we need to establish a proof for q using facts and strict rules. The deduction is in the classical sense - no proof of the negation of the q can be considered. The deduction shows that if q is proved at that (i+1)th layer then there exists a rule in the former layers that (till i) subsequently proves the reasoning in the account. In defeasible reasoning we must take account of opposing chains too.

$$
\begin{aligned}
&-\Delta: \text{If } P(i+1) = -\Delta q \text{ then}\\
&\qquad q \notin F \text{ and}\\
&\qquad \forall r \in R_x[q] \exists a \in A(r): -\Delta a \in P(1..i)
\end{aligned}
$$

The same reasoning as before, to prove −*Δq* then first q must not be a fact. In addition, we need to establish that every strict rule with head q is *known to be inapplicable.* Thus, for every rule such r there must be at least such rule *r* there must be at least one antecedent a for which we have established that a is not definitely provable (−Δa).

Now we move on to the provability of defeasible +∂q and -∂q

$$
\begin{aligned}
&+\partial: \text{If } P(i+1) = +\partial q \text{ then either}\\
&\quad (1) + \Delta q \in P(1..i) \text{ or}\\
&\quad (2) \quad (2.1) \exists r \in R_{nd}[q] \forall a \in A(r): +\partial a \in P(1..i) \text{ and}\\
&\qquad\quad (2.2) - \Delta \sim q \in P(1..i) \text{ and}\\
&\qquad\quad (2.3) \forall s \in R[\sim q] \text{ either}\\
&\qquad\qquad (2.3.1) \exists a \in A(s): -\partial a \in P(1..i) \text{ or}\\
&\qquad\qquad (2.3.2) \exists t \in R_{sd}[q] \text{ such that}\\
&\qquad\qquad \forall a \in A(t): +\partial a \in P(1..i) \text{ and } t > s
\end{aligned}
$$

Let us illustrate this definition. 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. 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”, i.e., reasoning chains in support of ∼q. To be more specific: to prove q defeasibly we must show that ∼q 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

~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 Section 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.

Similarly, the rule follows for -∂q.

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

Proposition 1 *If D is decisive, then for each literal p:*

1) *either D ⊢ +Δq or D ⊢ -Δq*
2) *either D ⊢ +∂q or D ⊢ -∂q*

Not every defeasible theory satisfies this property.

Strong negation principle

The principle of strong negation defines the relation between positive and negative conclusions. It preserves the coherence and consistency of the calculations derived. As shown in the proof theory, a negative conclusion can only be concluded when all positive counterparts are not derivable, i.e., for a literal q, the rules are defined in such a way that all the possibilities for proving +∂q are explored and shown to fail that -∂q can be concluded. The structure of the proof conditions is the same, but the conditions are negated in some sense. We say that the proof condition for a tag is the strong negation for its complement. That is, +∂ (or respectively −∂) is the strong negation of −∂ (or +∂). Table 2.1 below defines the formulas for strong negation (sneg)

sneg (+∂p ∈ X) = −∂p ∈ X

sneg (−∂p ∈ X) = +∂p ∈ X

sneg (A ∧ B) = sneg(A) ∨ sneg(B)

sneg (A ∨ B) = sneg(A) ∧ sneg(B)

sneg(∃xA) = ∀x sneg(A)

sneg(∀xA) = ∃x sneg(A)

sneg(¬A) = ¬sneg(A)

sneg(A) = ¬A where A is a pure formula

Coherence and consistency

In order to prove that the defeasible logic follows the property of classical logic. We know that it should follow the rules of coherence and consistency. A theory is coherent if, for a literal p, we cannot establish simultaneously that p is both provable or unprovable. That is, we cannot derive from the theory that $D \vdash +\Delta p$ *and* $D \vdash -\Delta p$, *or* $D \vdash +\partial p$ *and* $D \vdash -\partial p$.

On the other hand, consistency says that a literal and its negation can both be defeasible provable only when it and its negation are definitely provable; hence defeasible reasoning does not introduce inconsistency. A logic is coherent (consistent) if the meaning of each theory of the logic, when expressed as an extension, is coherent (consistent).

The elements of a derivation P in D are called lines of derivation. In the above definition often, we refer to the fact that a rule is currently applicable. This may create the wrong impression that this applicability may change as the proof proceeds (something found often in non-monotonic proofs). But the sceptical nature of DL does not allow for such situation.

For example, if we have established that a rule is currently not applicable because we have $-\partial a$ for some antecedent a, this means that we have proven at a previous stage that a is not provable from the defeasible theory D per se. The rule then cannot become applicable at a later stage of the proof or, indeed, at any stage of any proof based on the same defeasible theory.

Conversion of defeasible theory to Regular form

Definition 5.2. Consider a defeasible theory D = (F, R, >), and let Σ be the language of D. We define regular(D) = (∅, R′, >), where R′ is defined below. Let ′be a function which maps propositions to new (previously unused) propositions, and rule names to new rule names.

We extend this, in the obvious way, to literals and conjunctions of literals.

R′=Rd ∪ Rdft ∪

{→ f 0 | f ∈ F} ∪

{r′: A 0 → C 0 | r: A → C is a strict rule in R} ∪

{r: A ⇒ C | r: A → C is a strict rule in R} ∪

{p′→ p | A → p ∈ R or p ∈ F}. The rules derived from F and rules p′ → p is given distinct new names

A more formal precise approach to defeasible logic (via Non monotonic logic)

The intuition behind the classical logic is that it is based on such situations where we can derive any consequences. A more cautious approach is to consider maximal consistent subsets (with respect to set inclusion) of the given information (Rescher and Manor 1970). For instance, where p, q, and s are logical atoms and $\Gamma = \{p \wedge q, \neg p \wedge q, s\}$, maximal consistent subsets of Γ are $\Gamma 1 = \{p \wedge q, s\}$ and $\Gamma 2 = \{\neg p \wedge q, s\}$

Let Σ be a possibly inconsistent set of formulas and let MCS (Σ) be the set of maximal consistent subsets of Σ. The set Free (Σ) of *innocent bystanders in* Σ is obtained by intersecting all members of MCS (Σ).

- **Free consequences**. ϕ is a free consequence of Σ, denoted by Σ|∼freeϕ, if and only if it is (classically) entailed by the set of all the innocent bystanders Free (Σ).
- **Inevitable consequence**. ϕ is an inevitable consequence of Σ, denoted by Σ|∼ieϕ, if and only if it is (classically) entailed by each member of MCS (Σ).

The classical logic follows the principle of monotony i.e.

Monotony: If $\Sigma|\sim\phi$ then also $\Sigma\cup\Sigma'|\sim\phi$.

Monotony states that consequences are robust under the addition of information: if ϕ is a consequence of Σ then it is also a consequence of any set containing Σ as a subset.

The formal properties which are replaced instead of monotony is as follows: -

- **Cautious Monotony**: If $\Sigma|\sim\psi$ and $\Sigma|\sim\phi$, then also $\Sigma\cup\{\phi\}|\sim\psi$.
- **Rational Monotony**: If $\Sigma|\sim\psi$, and $\Sigma/\sim\neg\phi$, then also $\Sigma\cup\{\phi\}|\sim\psi$.

Both Cautious and Rational Monotony are special cases of Monotony, and are therefore not in the foreground as long as we restrict ourselves to the classical consequence relation $\vDash$ of CL

In CL a formula φ is entailed by Γ (in signs $\Gamma \vDash \varphi$) if and only if φ is valid in all classical models of Γ. An influential idea in NML is to define non-monotonic entailment not in terms of *all* classical models of Γ, but rather in terms of a *selection* of these models (Shoham 1987). Intuitively the idea is to read

$\Gamma|\sim\varphi$ as "φ holds in the most normal/natural/etc. models of Γ."

Preferential structures enjoy a central role in NML since they characterize *preferential consequence relations*, i.e., non-monotonic consequence relations $|\sim$ that fulfil the following central properties, also referred to as the *core properties* or the *conservative core* of non-monotonic reasoning systems or as the *KLM-properties* (in reference to the authors of Kraus, Lehmann, Magidor 1990):

Reflexivity: $\varphi|\sim\varphi$.

Cut: If $\varphi\wedge\psi|\sim\tau$ and $\varphi|\sim\psi$, then $\varphi|\sim\tau$.

Cautious Monotony: If $\varphi|\sim\psi$ and $\varphi|\sim\tau$ then $\varphi\wedge\psi|\sim\tau$.

Left Logical Equivalence: If $\vDash\varphi\equiv\psi$ and $\varphi|\sim\tau$, then $\psi|\sim\tau$.

Right Weakening: If $\vDash\varphi\supset\psi$ and $\tau|\sim\varphi$, then $\tau|\sim\psi$.

OR: If $\varphi|\sim\psi$ and $\tau|\sim\psi$, then $\varphi \vee \tau|\sim\psi$.

According to Left Logical Equivalence, classically equivalent formulas have the same non-monotonic consequences. Where ψ is classically entailed by φ, Right Weakening expresses that whenever φ is a no

AND: If $\varphi|\sim\psi$ and $\varphi|\sim\tau$ then $\varphi|\sim\psi\wedge\tau$.

S: If $\varphi\wedge\psi|\sim\tau$ then $\varphi|\sim\psi\supset\tau$

Example discussion of defeasible logic through Spindle

We would like to implement our defeasible reasoner on the various exception laws. Spindle is a Java API tool which creates each class instance for the reasoning. Further detailed case study of spindle will be done later in the report. Henceforth, let us look at the ACT health act of 1993, particularly the subdivision of abortion. The main idea for focusing on this subsection is that we can allow the process of having **exceptions** is our knowledge base. Moreover, allowing exceptions can help us to identify that the given set of rules are consistent with our current set of laws. The formalisation of the abortion law would have defeasible rules in order to allow the exceptions. Let us formally see the set of rules for our knowledge base: -

Facts >>person(supply)

>>purpose(endingpregnancy)

>>assisting(pharmacist)

>>pharmacist(supply)

Defeasible rules

r1: person(supply), purpose(endingpregnancy), -doctor => abortion(offence)

r2: assisting(pharmacist), pharmacist(supply) => -abortion(offence)

r3: abortion(offence) => guilty

r4: =>-guilty

The first rule states that if a person supply an abortifacient(drug for abortion) to another person and he/she has the sole purpose of ending pregnancy and the person is not a doctor then we can defeasibly imply that he/she has committed an offence. Since the rule r1 can have exceptions, we need to use defeasible implication. The second rule is the exemptions from the first rule i.e. If the pharmacist is supplying the abortifacient (in accordance with prescription) or a person is assisting the pharmacist then he/she has not done the abortion offence.

The above set of rules (i.e. r1 and r2) can be presumed as a set of evidence indicating certain offence and both of them are reliable (i.e. facts). Henceforth, the literals such as, the person who is supplying, pharmacist(supply) and a person who is assisting the pharmacist are all facts and their provability is not questionable. The rule r3 states that if someone has committed an offence then it is guilty. Furthermore, the rule r4 states that according to the underlying legal system a defendant is presumed innocent (i.e., not guilty) unless responsibility has been proved (without any reasonable doubt).

Now in order to check our knowledge base, let us check the above set of rules with an inconsistent rule. In natural language the rule is described as follows, If a person is a doctor and he/she has the sole purpose of ending the pregnancy then he/she should be proven guilty or he/she must have a committed an offence.

doctor, purpose(endingpregnancy) => abortion(offence)

Note that in our current defeasible rules we are not defining any superiority rules because every other rule is specific enough to conclude the fact that an offence is committed or not. If we had a more specific antecedent, then we could have a superiority relation. Henceforth the given set of rules will solve the process as ambiguity propagation in non-monotonic logic.

Let us examine the results from our spindle reasoner: -

Conclusion	
+D	assisting(pharmacist)
+D	person(supply)
+D	pharmacist(supply)
+D	purpose(endingpregnancy)
-D	abortion(offence)
-D	-abortion(offence)
-D	doctor(X)
-D	-doctor(X)
-D	guilty(X)
-D	-guilty(X)

+d	**-abortion(offence)**
+d	assisting(pharmacist)
+d	**-guilty(X)**
+d	person(supply)
+d	pharmacist(supply)
+d	purpose(endingpregnancy)
-d	abortion(offence)
-d	doctor(X)
-d	-doctor(X)
-d	guilty(X)

1) The first four literals were facts; hence they are definitely provable in our reasoning. The next six literals show that the literals which are not facts are not definitely provable and it can be examined by looking the set of rules, we have literals (abortion(offence) and -abortion(offence)).

2) The literals which were facts i.e. (assisting(pharmacist), person(supply) and purpose(endingpregnancy) are also defeasibly proved +d.

3) Now focus on the literals **-abortion(offence)** and **-guilty(X)** these two literals show that we can defeasibly prove that the person has not committed an offence and he/she has been presumably defined as not guilty. Moreover, if not specifying any superiority rule, we can see that the person who is a doctor and has a sole purpose of ending the pregnancy can be proven as not guilty with current set of laws. The last set of literals also showcase that abortion(offence) is not defeasibly proved and guilty is not defeasibly proved as well.

Extending the defeasible logic with modal operators

Usually modal logics are extensions of classical propositional logic with some intentional operators. Thus any modal logic should account for two components: (1) the underlying logical structure of the propositional base; and (2) the logical behaviour of the modal operators. The classical logic are not well suited for real life scenarios.

Narrowly construed, modal logic studies reasoning that involves the use of the expressions ‘necessarily’ and ‘possibly’

Logic	Symbols	Expressions Symbolized
Modal Logic	□	It is necessary that …
	◊	It is possible that …
Deontic Logic	O	It is obligatory that …
	P	It is permitted that …
	F	It is forbidden that …

The most familiar logics in the modal family are constructed from a weak logic called **K** (after Saul Kripke). A variety of different systems may be developed for such logics using **K** as a foundation. The symbols of K include '∼' for 'not', '→' for 'if…then', and '□' for the modal operator 'it is necessary that'. (The connectives '&', 'V', and '↔' may be defined from '∼' and '→' as is done in propositional logic.)

$\Diamond A = \sim\Box\sim A$

Distribution Axiom: $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$

The Distribution Axiom says that if it is necessary that if A then B, then if necessarily A, then necessarily B.

The modal operators act just like quantifiers and existential in predicate logic, such as $\Box(A\&B)$ entails $\Box A \& \Box B$ and vice versa.

$$\Box A \rightarrow \Box\Box A \qquad (4)$$

$$\Diamond A \rightarrow \Box\Diamond A \qquad (5)$$

In S4, the sentence $\Box\Box A$ is equivalent to $\Box A$. As a result, any string of boxes may be replaced by a single box, and the same goes for strings of diamonds. This amounts to the idea that iteration of the modal operators is superfluous. Saying that A is necessarily necessary is considered a uselessly long-winded way of saying that A is necessary. The system S5S5 has even stronger principles for simplifying strings of modal operators. In S4, a string of operators of *the same kind* can be replaced for that operator; in S5, strings containing both boxes and diamonds are equivalent to the last operator in the string. So, for example, saying that it is possible that A is necessary is the same as saying that A is necessary.

Deontic logic operators

Deontic logics introduce the primitive symbol O for 'it is obligatory that', from which symbols P for 'it is permitted that' and F for 'it is forbidden that' are defined: $PA = \sim O \sim A$ and $FA = O \sim A$

For the process of allowing the modal operators in the defeasible logic, we must use the proper notation allowing us to derive special representation in every form. For example, the deontic statement, "The Purchaser shall follow the Supplier price lists" can be represented as

$$\text{AdvertisedPrice}(X) \Rightarrow_{O_{\text{purchaser}}} \text{Pay}(X)$$

The following notation has been used throughout the paper for modal defeasible logic

The language of Modal Defeasible Logic (MDL) consists of a finite set of modal operators Mod = {□1, . . . , □n} and a (numerable) set of atomic propositions Prop = {p, q, . . . } 6 . Besides,

The definition of literal (an atomic proposition or the negation of it) is supplemented with the following clause: • If l is a literal then $\Box_i l$, and $\neg\Box_i l$, are literals if l is different from $\Box_{im}$, and $\neg\Box_{im}$, for some literal m. The basic intuition that follows the derivability of modal operators is that if we can drive p then we can build the derivation of $\Box p$. By building up the same intuition, we want to replace the derivability of classical logic with the practical and feasible derivability in defeasible logic. The main idea is to allow to derive $\Box p$ if we can prove p with mode □ in Defeasible logic. The simple treatment of modalities: what we obtain is that the conditions for introducing modalities (intentions and obligations) collapse into those for deriving literals in standard Defeasible logic.

Rule conversion

The notation of rule conversion allows us to model peculiar interactions between different modal operators. We will explore the conversion of modal operators using the *right weakening* rule expressed in the formalism via non monotonic logic.

$$\frac{B \mid\!- C \qquad A \mid\!\sim B}{A \mid\!\sim C}$$

Suppose a rule of the specific type is given and suppose all the literals in the antecedent of a rule are provable in same modality. Suppose one strong modal conclusion inherits the modality of another rule as antecedent. The above conversion can be represented as follows: -

$$\frac{\Gamma \quad \mid\sim \mathrm{INT}\psi \quad \psi \Rightarrow_{\mathrm{BEL}} \phi}{\Gamma, \mathrm{INT}\psi \quad \mid\sim \mathrm{INT}\phi}$$

Let us clear this conversion with an example, if an agent **believes** to visit Italy(phi) if she visits Rome(psi), and she has the intention to visit Rome, then it seems rational that she has the intention to visit Italy. Consider another example of the formalism

Load_live_ammo,shoot => $_{BEL}$ *kill*

This rule encodes the knowledge of an agent that knows that loading the gun with live ammunitions, and then shooting will kill her friend. This example clearly shows that the qualification of the conclusions depends on the modalities relative to the individual act's "load" and "shoot". In particular, if we obtain that the agent intends to load and to shoot the gun (INT(load), INT(shoot)), then, since she knows that the consequence of these actions is the death of her friend, she intends to kill him. However, if shooting was not intended, then we have prima facie to say that killing, too, was not intentional.

Conflict resolution

Consider the following example for little bit more clarification: -

$$r: a \Rightarrow_{BEL} q, \quad s: b \Rightarrow_{BEL} -q, \quad t: c \Rightarrow_{INT} q$$

If we have only Conflict(BEL,OBL), this means that rule r conflicts with rule s that r is stronger than s : for this reason, if applicable, r will defeat s. Now consider only rule s and rule t. There is no incompatibility relation between INT and OBL and we are free to derive both INT_q and OBL_{-q}.

The language of Modal Defeasible logic

The inference process consists of factual knowledge, intentions and obligations based on existing knowledge base. Provability for beliefs does not generate modalized literals, since in our view beliefs concern the knowledge an agent has about the world and corresponds to the basic inference mechanism of the agent.

Definition 1 (Language) Let PROP be a set of propositional atoms, MOD = {BEL, INT, OBL} be the set of modal operators, and lab be set of the labels. The sets below are the smallest sets closed under the following rules:

Literals

Lit = PROP U {-p|p ∈ PROP}

If q is a literal, ~q denotes the complimentary literal (if q is positive literal p then ~q is -p; and if q is -p, then ~q is p).

Modal literals

ModLit = { XL,-XL|L ∈ Lit , X ∈ {INT,OBL}}

Rules = Rule_s *U* Rule_d *U* Rule_{dft}, where for X ∈ MOD

$$\begin{aligned}\text{Rule}_s = \{r : \phi_1, \ldots, \phi_n \to_X \psi | \\ r \in \text{Lab}, A(r) \subseteq \text{Lit} \cup \text{ModLit}, \psi \in \text{Lit}\} \\ \text{Rule}_d = \{r : \phi_1, \ldots, \phi_n \Rightarrow_X \psi | \\ r \in \text{Lab}, A(r) \subseteq \text{Lit} \cup \text{ModLit}, \psi \in \text{Lit}\} \\ \text{Rule}_{dft} = \{r : \phi \leadsto_X \psi | \\ r \in \text{Lab}, A(r) \subseteq \text{Lit} \cup \text{ModLit}, \psi \in \text{Lit}\}\end{aligned}$$

$$\begin{aligned}\text{Rule}^{\text{BEL}} = \{r : \phi_1, \ldots, \phi_n \rhd_{\text{BEL}} \psi | \\ (r : \phi_1, \ldots, \phi_n \rhd_{\text{BEL}} \psi) \in \text{Rule}, \rhd \in \{\to, \Rightarrow, \leadsto\}\} \\ \text{Rule}_s[\psi] = \{\phi_1, \ldots, \phi_n \to_X \psi | \\ \{\phi_1, \ldots, \phi_n\} \subseteq \text{Lit} \cup \text{ModLit}, \psi \in \text{Lit}, X \in \text{MOD}\}\end{aligned}$$

Given a defeasible agent theory D, +ΔXq means that literal q is provable in D using only facts and strict rules for modality X, −ΔXq means that it has been proved in D that q is not definitely provable in D, +∂Xq means that q is defeasibly provable in D, and −∂Xq means that it has been proved in D that q is not defeasibly provable in D.

Example 1 (Running example). Frodo, our Tolkienian agent, is entrusted by Elrond to be the bearer of the ring of power, a ring forged by the dark lord Sauron. Frodo has the task to bring the ring to Mordor, the realm of Sauron, and to destroy it by throwing it into the fires of Mount Doom. However, Frodo loves the place where he was born, the Shire, and intends to go there.

F = {INTGoToShire, EntrustedByElrond}

R = {r1 : EntrustedByElrond ⇒BEL RingBearer

r2 : RingBearer ⇒OBL DestroyRing

r3 : INTGoToShire ⇒INT ¬GoToMordor

r4 : ¬GoToMordor ⇒BEL ¬DestroyRing}

>= {r4 > r2}

C = {Convert(BEL,INT)}

V = {Conflict(BEL, OBL)}

Below is the set C of all conclusions we get using the rules in R: C = {RingBearer , INT¬GoToMordor , INT¬DestroyRing} As facts, we know that Frodo has the primitive intention to go to the Shire and that he has been entrusted by Elrond. These facts make applicable rules r3 and r1, which permit to derive that Frodo is the ring bearer and that he has the intention not to go to Mordor. At this point we have a conflict, as we have Conflict(BEL, OBL) and Convert(BEL,INT). In effect, given the conversion, r4 permits to derive that Frodo has the intention not to destroy the ring while rule r2 should lead to the obligation to destroy it. However, r4 is stronger than r2 and so we only get +∂INT¬DestroyRing.