2.5: Application of defeasible logic in various real-world scenarios

The ability of allowing the exceptions in our knowledge base helps in understanding the applications of all these formalization in real-world scenarios. This section will primarily focus on applying the laws of defeasible logic on various subsection of legislatives acts and we will briefly touch on why this kind of reasoning is ideal for some scenarios, while some may not oblige to our intuition.

2.5.1 : Australian capital territory health act 1993

We will be primarily focus on a subdivision of the part 5 in the act known as abortions. The goal of the formalization is to capture any inconsistency in the set of laws, and we know that one can capture those inconsistency by allowing them in our current knowledge base and argue it's provability by processing them in our inference engine.
Let us get acquainted with some of formal terms in this subdivision: -
Abortifacient means a medicine, drug or other substance that causes a pregnancy to end prematurely.
Abortion means a medical abortion or surgical abortion.
Let us look at the laws in form of defeasible logic

r1: personsupply, purposeendingpregnancy, -doctor => abortion(offence)
r2: assistingpharmacist, pharmacistsupply => -abortion(offence)
r3: abortion(offence) => guilty
r4: =>-guilty

The first rule states that if a person supply an abortifacient 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.

r5: doctor, purposeendingpregnancy => 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. After processing these rules in our inference engine, we got the derivation of tagged literals. The main highlight of the results were that we derived the +d -*abortion(offence)* and +d -*guilty(x)*, which showcase that we can defeasibily prove that abortion offence was not committed and the presumption of innocence is proved.
The above situation illustrates that there are two set of evidence which supports the conclusion of the abortion offence (rule r1 and rule r2) and after introducing the inconsistency rule in the knowledge

base there is another set of evidence (i.e. rule) that supports some specific conclusion. Furthermore, we have not specified any superiority rule in this whole knowledge base, hence the problem is formed as *ambiguity propagation* in defeasible logic. The situation arises when we have a set of evidence concluding an argument and then another set of evidence concluding the negation of same argument. The ambiguity could be solved but it is outside the scope of this current project.

The above formalization could have been solved without ambiguity, if in the underlying set of rules did have same sort of superiority between them. Henceforth it is important to note that the manifestation of defeasible rule to any formal system must be done by carefully supervising the analogy of those underlying laws to defeasible theory. Moreover, these formalizations cannot be applied to any set of laws, first they must oblige to our main reason i.e. to allow exceptions in the domains of knowledge base, secondly those exceptions can be defeated by more specific law that overrides the other.

2.5.2 : Stronger Futures in the Northern Territory Act 2012

In this section we will primarily focus on license regime of community stores established by the northern territory state. The licensing regime for community stores applies to stores in all the prescribed areas of the Northern territory. Mainly we will cover the stores which play an important role in assisting the food security of Aboriginal people. The determination of license on these community stores is an important step in safeguarding the people which inhibit these lands for decades. Hence, we will primarily focus on the rules that play a key role in determination of these license and after that we would argue that current formalization of the rules is strong enough to support any inconsistencies.

First, we will look the subject matter as a process of a natural language, then we will formalize the laws. The secretary of the state determines which store is balanced enough to hold a community store license. The conditions that are required for determination is 1) the store should be financially stable 2) It should be a rich source of food and grocery to the nearby community. Furthermore, the license can not be obtained if the owner and manager of the store has not given a written notice to the secretory underlying the reason for the license. The written notice must have a reason for obtaining the license, and a short summary from both owner and manager. The written notice must be submitted in definite period of time after lodging the application unless the secretory has agreed for longer period of time. The license would not be issued if the documents provided by the owner or manager are forged. Now if we formulate the above rules in defeasible theory then we get,

R1: financialstability(x) => determination(license)
R2:sourceoffood(x),sourceofgrocery(x) => determination(license)
R3: Notice(owner),Notice(manager)=> determination(license)
R4:reason(x),writtensummary(x)=>Notice(owner)
R5:reason(x),writtensummary(x)=> Notice(manager)

Here the predicate financialstability(x) or sourceoffood(x) denotes the whole literal and x can be replaced by any ground atom i.e. name of any aboriginal store. The above set of rules seems reasonable for defeasible rules perspective, but you cannot formulate the rules which have reparation. For example, if the notice is not submitted within given time frame then the one cannot get the license. Similarly, if all these conditions are not fulfilled then as a result the license would not be obtained. Furthermore, if we allow the inconsistency rule in the knowledge base, i.e. what if the manager and the owner are the same person and they are forging the documents in such a manner that it seems different person (but in actual are same), so that does illustrates that they can obtain the license. Hence formulating these types of inconsistency can be hard in simple defeasible logic. Furthermore, if we manage to formalize the inconsistency and allow them to reason with our knowledge base it will be a bunch of literals whose provability is in question. For example, which literals should be posed as facts, because anything could be forged in any manner.

There is no obligation or reparation in the rules provided above, hence we need something different type notation and reasoning. The defeasible logic can be extended 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 behavior of the modal operators. The classical logic is not well suited for real life scenarios.

We will not go in detail of modal extension with defeasible logic, interested reader can refer [8]. Just small example to clarify the operator use,

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

Here the above example illustrates that, "The purchaser shall follow the Supplier price lists". The primitive symbol O is defined as 'it is obligatory that'. Now we can reformulate the above laws as extension with modal logic

R1: $O_{s,b}$*financialstability(x),* $O_{s,b}$*sourceoffood(x),* $O_{s,b}$*sourceofgrocery(x) => determination(License)*

R2: $O_{s,b}$*Notice(owner),* $O_{s,b}$*Notice(manager) => determination(License)*

Here $O_{s,b}\,A$ states that A is obligatory such that s is subject of obligation and b is the beneficiary. Moreover, the subject of obligation in this case is determination of license and store owner and manager are the beneficiary.

The above rules can be summed up to one rule as shown below

R3: determination(license) =>$\otimes_{i=1}^{n} O_{s,b} B_i$ ⊗ ¬determination(license)

The symbol ⊗ denotes reparation, the above equation demonstrates that the determination of license is subjected to the above set of obligations and if any one of them is not succeeded then the reparation is the failure to obtain the license. Henceforth, we formulated all these rules under the same modal extension of defeasible logic. However, we could not formulate the inconsistency due to its ambiguous nature. Moreover, the reasoning of deontic defeasible logic is complicated which requires extensive work. However, the above approach gave us that how we can formalize certain laws that are in a sense of obligation and reparation.

2.5.3 : Final exam timetabling of a university

The following example will exploit the nature of final exam and the related course work period in the university. This example can showcase that for a simple defeasible rule we can have multitude of exceptions and the superiority between these exceptions can be hierarchical [9]. The example will showcase an easy modification of natural language into defeasible logic. Moreover, the results exploited by the defeasible theory reasoner can be sustained by balancing some exceptions than the other.

Let us describe the natural language of all these rules:-

1) The students who are preparing for the final exam in the university usually have one-week preparation time unless the final exam content is not a big part in their coursework. Moreover, there should be at least a week gap between the final exam and the end of course content.

2) The one-week duration of the final exam should be in week 13 followed by the commencement of final exam by end of week 13 and going on till week 14 and week 15. Now the courses which do not have final exam (i.e. have continuous assessment scheme) does not have to ordinarily follow the guidelines in the subsection(1). Moreover, for those courses the teaching period can be extended by the week 13.
3) If any of the course lecturer seems that the teaching period is extended beyond the week 13 (because of the constraint regarding the public holidays) then it should be the case that the revision week must be shifted accordingly and then the whole final exam timetabling schedule.

The above formulation of the rules can be done using simple defeasible logic without any modal or deontic extensions. Then we will discuss how the rule 1 and rule 3 seems to have a little bit of redundancy in the formulation which is also known as anomaly detection.

R1: finalcontent(x),oneweekduration => commencementexam(x)
R2: finalexampercentagelessthan40 => ¬oneweekduration
R3: oneweekduration => week13
R4: finalexampercentagemorethan40 => oneweekduration
R5: continousassesment(x) => ¬commencementexam(x)
R6: oneweekduration, continousassesment(x) => ¬week13
R7: lectureragreesonextension(x) => ¬commencementexam(x)

The rule R7 is superior then rule R1 because it has hierarchical advantage over these rules. It is clearer in introspection than any other rule. Furthermore, let us simply examine the above set of rules with a course (example comp 4450) this course is a continuous assessment.
The rule R5 would be as follows
continousassesment(comp4450) => ¬commencement(comp4450) and
The rule R6 is as follows
Oneweekduration, continousassesment(comp4450) => ¬week12
This is where the hierarchy of the superiority rule takes place if we process these rules in SPINDLE with R5 > R1 we would obtain the result as *+d ¬commencementexam(comp4450)*. The implicit constraints about the course comp 4450 would be that its is indeed a continuous assessment and it is a fact. Hence, the provability is undisputable. Furthermore, these implicit constraints can sometime help us in facilitating the superiority between the rules. Moreover if we had another example course such as comp 3620 which is not a continuous assessment course and which is indeed a fact, then the rule R1 would override other rule such as R5, and derive the proof tagged literal as *+d commencementexam(comp3620).*
Hence, the superiority between the rules could be a process of regulating the rules. It is possible to have specific result desired to user's own regulation.

2.6: Brief discussion regarding the application

The formal representation of the above examples can be a process of drafting specific rules or regulation according to user's specific choice. In the first example we formed the rules for a subset of laws in the health care act and the result were the process of ambiguity propagation in defeasible logic. The process can be seen as *Anomaly detection* i.e. before formalizing the problem we did not know that there would be any circular dependency or ambiguous results. Hence, this process could be understood as checking any formal system for any inconsistencies and since we are not specifying any superiority rules , we let the reasoner to lay out the proof. The result would have been very different if we allowed priority of inconsistency rule in defeasible theory. The second example can be seen as the matter of *hypothetical testing* since we formalized the set of rules which had rather large domain. Moreover, the formalization was done through modal extension, we hypothesized over an inconsistency that what would be the case that if owner and manager are the same person and they have forged the documents for the license. Hence, we could not get the answers because there was an ambiguity on how to allow those rules in deontic logic. The third example focused on the aspect of *debugging* i.e. in many of those cases we know that what actual answer of the query would be, as the course comp 4450 is continuous assessment course , it would have greater precedence over the other rules(because of the hierarchical structure). Hence if we change current set of regulation (superiority relation) it would give us a different result. Debugging suggests that changes to the regulations which will have an effect on the desired outcome.