Project on *The Rational Behind the Concept of a Goal*

Hannah Yao

2019 December

1 Overview

The Rationale Behind the Concept of a Goal by Guido Governatori et al. [2] aims to formalize into a set of rules the decision process of rational and cognitive agent operating in an environment with some objectives to achieve given the existence of personal desires and contextual obligations. The system is abstracted and represented as, (1) the environment in which the agent exists, and how the agent views the world (2) the norms and constraints of the application domain, and (3) the agent's internal constraints and objectives.

The abstraction is based on the *Belief, Obligation, Intention, Desire* model, in which mental attitude embodies the objectives and beliefs describe the environment (through one's own view of the world), and there may be conflicts between the agent's desires and obligations or the norms. The paper purposes that: goals, desires, and intentions are all part of the same phenomenon, and unified into the notion of an outcome, which is something that the agent can expect to ultimately happen (end goal). In addition, the agent prefers certain outcomes over others. Thus the agent's wants are affected by obligations and norms, which results in a subset of actions that is the outcome.

Defeasible Reasoning
Reasoning is defeasible when the corresponding argument is rationally compelling but not deductively valid. The truth of the premises of a good defeasible argument provide support for the conclusion, even though it is possible for the premises to be true and the conclusion false. In other words, the relationship of support between premises and conclusion is a tentative one, or modal [3]. Defeasible reasoning can be studied as a branch of logic through a argumentative way, which represents a way of reasoning over a knowledge base containing possibly incomplete and/or inconsistent information, to obtain useful conclusions, and Defeasible Logic Programming offers a computational reasoning system that uses an argumentation engine to obtain answers from a knowledge base represented using a logic programming language extended with defeasible rules [1].

Project Objective
Python implementation of Defeasible Extension Algorithm as mentioned in the paper, which computes the defeasible conclusions for given modality (not complete/functional currently, but complete enough to be read as psuedocode).

2 Definitions and framework

The agent's knowledge base is represented as rules, for example:

$$r_1 : drive\_car \Rightarrow_o \neg damage \oplus compensate \oplus foreclosure \tag{1}$$

says that: if an agent drives a car, then he has the obligation not to cause any damage; if he causes damage, then he is obliged to compensate; if he fails to compensate, then there is an obligation of foreclosure. $\oplus$ is

a preference operator. This illustrates a sequences of outcomes or alternatives that are ordered from most to least desirable. All knowledge (derivable) either exist as rules, or as a Fact (the knowledge base). The paper formally defines the following:

1. Desires are acceptable outcomes and always compatible with other expected or acceptable outcomes. If $b1 = \neg b2$, both b1 and b2 are acceptable (2.1). However, if $\neg b1 \in Knowledge\_Base$, then +b1 must be discarded; else this is wishful thinking and not rational.

2. Suppose $r1 \Rightarrow visit$ and $r2 \Rightarrow \neg visit$, and we know, for example, Alice prefers to visit John (r1) but that she knows she should not visit John because she has to do homework (r2), but she believes r1 >r2 (is superior). Then the elements of the weaker rule with an incompatible counterpart in the stronger rule are not considered desires (2.1, 2.2). And Alice would visit John.

3. Goals are preferred outcomes in a chain of alternative choices. Using rule superiority, outcomes can be discarded for the weaker rule (2.2). Thus a goal is the most preferred outcome.

4. The agent's beliefs can determine what is achievable. If a rational agent, for example, believes $\neg visit$ holds, then attempting *visit* is a waste of effort and should focus on the next best outcome. Otherwise the agent is not rational (2.3).

5. The "next best outcome" mentioned above is regarded as *intention* (2.3). An intention is an acceptable outcome that does not conflict with the agent's internal beliefs about the environment.

6. A *social intention* is an intention that does not violate societal or external norms. For example, if Alice wants to visit John, but John is under house arrest, then even though she has the intention, she would have to resort to another outcome (social intention) because the law enforces she cannot visit John ($\neg visit$ holds).

7. Unit of interest is a *modal operator applied to a liter*: MOD = {B,O,D,G,I,SI} for belief, obligation, desire, goal, intention, social intention. And MODLIT = {Xl, $\neg$ Xl | l is a propositional atom that is p or $\neg$p, and X $\in$ MOD-{B}}. A belief *Bl* is equivalent to the single literal *l* since beliefs describe environment (3 def. 1).

8. A *defeasible theory D* is a structure (Facts F, Rules, >) where $F \subseteq Literal \cup MODLIT$ and is a set of indisputable statements. *R* consists of three different types of rules for beliefs, obligations, and outcomes. And > is for resolving conflict between rules (3 def. 2). For example, O$\neg visit \in F$ means there exists the obligation to fulfill outcome not visit.

9. *Belief rules* relate factual knowledge of an agent and relate states: The world as seen from the agent's perspective.

10. *Obligation rules* determine which obligations are active and should be met.

11. *Outcome rules* enumerate possible outcomes depending on the context. Outcome and obligation rules derive conclusions for all modes representing goal-like attitudes, which includes desires, goals, intentions and social intentions.

The following briefly state rule applicability and defeasible provability in English.

1. *Defeasible rule* is an expression: r: A(r) $\Rightarrow_X$ C(r) where r is the name of the rule, A(r) = $\{a_1...a_n\}$ and a set of premises, ordered alternatives. $X \in \{B, O, U\}$, and C(r) is the consequent or head of the rule. It is either a single literal if X = B or a preference expression. For example, A(r) $\Rightarrow_o c$ and the premises hold, then *r* can be used to prove Oc (3 def. 3).

2. *Body applicable* : A rule $r \in R$ is body-applicable iff for a Proof P and X $\in \{O, D, G, I, SI\}$ for all $a_i \in A(r)$ such that (1) $a_i = Xland - \delta_X l \in P(1...n)$ (2) if $a_i = \neg Xl$ then $-\delta_X l \in P(1...n)$ (3) if $a_i = l \in Literal$ then $+\delta_B l \in P(1...n)$.

3. *Rule conversion* : Convert(X,Y) states rule of mode X can produce conclusions of mode Y. In addition, non-beliefs cannot convert to beliefs, but beliefs can convert to another mode, since having a goal cannot change how the agent initially viewed the world. Different modal operators interact through *rule conversion* and *rule resolution*.

4. *Conflict detection and resolution* : Conflict(X, Y) such that X and Y conflict and X >Y. Consider: Conflict(Belief, Intention), Conflict(Belief, Social Intention) and Conflict(Obligation, Social Intention). Conflict between belief and intention modes implies agents can recognize conflict and are therefore realistic. Conflict(O, SI) implies agent is aware of norms. Conflict is resolved by >.

5. A *Proof* consists of P(1)...P(n) of tagged literals that are denoted $+\delta_X q$ or $-\delta_X q$ where X ∈ MOD, $+\delta_X q$ means that consequent q is defeasibly provable in mode X in a given defeasible theory. If it is not provable then it is refuted, or in $-\delta_X q$ (def. 4).

6. Intuitively: Rules can be either applicable or discarded depending on the body of the rule. When considering obligations, if we are considering the ith element, then that is only because all previous elements are violated. So a literal can only be a goal if none before it has been proved as such. Intentions must have no factual knowledge of the opposite conclusion else wise it is wishful thinking (def. 8).

7. Example: $F = \{a, b, Oc\}$ and $R = \{r1 : a \Rightarrow_o b$, r2: $b, c \Rightarrow d\}$, r1 is applicable while r2 is not because c does not exist in F. But r2 is Convert-applicable because Oc is a fact, and r1 produced Ob (def. 8).

8. In order to assert *q* in an element chain as a desire, then there must be no stronger argument (or rule) of opposite desire.

9. *Defeasible provability for obligation, goal, intention and social intention*. In order to assert literal *q* is defeasibly provable in modality X: (1) Xq is a fact or (2) a complementary literal of same or conflictual modality does not appear as a fact and there is some applicable rule for X and q. (3) Every rule for case $\neg q$ of a same or conflicting modality must be discarded. (4) There is also no stronger attacking rule for case $\neg q$ (def.13).

10. Example: $F = \{a_1, a_2, \neg b_1, O\neg b_2\}$ and $R = \{r1 : a_1 \Rightarrow_u b_1 \oplus b_2 \oplus b_3 \oplus b_4, r2 : a_2 \Rightarrow_u b_4\}$. Where $\Rightarrow_u$ is the outcome rule. Every b_i is trivially a desire.
Consider we have as goal b_1, but $\neg b_1 \in F$ $(+\delta\neg b_1)$ so we cannot have that, which means $-\delta_{I,SI} b_1$. Then everything beyond b_1 is considered as intentions. Since b_2 is intention I then b_3, b_4 are discarded as intentions. But since O$\neg b2$ is a fact, r1 is discarded for SI and b2 which makes SI applicable again for SI and b3, proving $+\delta_{SI} b_3$. Then r1 is discarded for b4. But the defeasible theory still derives b4 because of r2, such that $Defeasible\_theory \vdash +\delta_{D,G,I,SI} b_4$.

11. In addition, if we consider literal q as a desire D, then the modality for q cannot be {G,I,SI} since literals are considered in preference order. If we discard q as desire, then we would move onto considering q as intention, which would be irrational because q was already shown as unachievable (proposition 3).

3 Computing the extension of a finite defeasible theory

An extension is the smallest set that contains all the facts of the theory. In this case it would be a set of outcomes given the environment constraints.

For the algorithm notation: ■ denotes a generic modality, □ denotes a fixed chosen modality in ■. R_{sup} is the set of superior rules such that r >s, and R_{inf} is the set of inferior rules. The Herbrand Base $HB = \{\Box l | \Box \in MOD, l \in HB_D\}$ where HB_D is the set of literals such that a literal or complement

appears in the defeasible theory. The consequents of rules are modified via truncation and or removal of alternatives. The *defeasible extension* is $E(D) = (+\delta_{\blacksquare}, -\delta_{\blacksquare})$. The algorithm uses operations of truncation

Algorithm 1 DEFEASIBLEEXTENSION

1: $+\partial_{\blacksquare}, \partial^{+}_{\blacksquare} \leftarrow \emptyset$; $-\partial_{\blacksquare}, \partial^{-}_{\blacksquare} \leftarrow \emptyset$
2: $R \leftarrow R \cup \{r^{\square} : A(r) \Rightarrow_{\square} C(r) | r \in R^{\mathsf{U}}\}$, with $\square \in \{\mathsf{D}, \mathsf{G}, \mathsf{I}, \mathsf{SI}\}$
3: $R \leftarrow R \setminus R^{\mathsf{U}}$
4: $R^{\mathsf{B},\diamond} \leftarrow \{r^{\diamond} : A(r) \hookrightarrow C(r) | r \in R^{\mathsf{B}}, A(r) \neq \emptyset, A(r) \subseteq \mathrm{Lit}\}$
5: $>\!\leftarrow\!> \cup \{(r^{\diamond}, s^{\diamond}) | r^{\diamond}, s^{\diamond} \in R^{\mathsf{B},\diamond}, r > s\} \cup \{(r, s) | r \in R^{\blacksquare} \cup R^{\mathsf{B},\blacksquare}, s \in R^{\diamond} \cup R^{\mathsf{B},\diamond}, \mathrm{Conflict}(\blacksquare, \diamond)\}$
6: **for** $l \in F$ **do**
7: **if** $l = \square m$ **then** PROVED(m, $\square$)
8: **if** $l = \neg\square m \wedge \square \neq \mathsf{D}$ **then** REFUTED(m, $\square$)
9: **end for**
10: $+\partial_{\blacksquare} \leftarrow +\partial_{\blacksquare} \cup \partial^{+}_{\blacksquare}$; $-\partial_{\blacksquare} \leftarrow -\partial_{\blacksquare} \cup \partial^{-}_{\blacksquare}$
11: $R_{infd} \leftarrow \emptyset$
12: **repeat**
13: $\partial^{+}_{\blacksquare} \leftarrow \emptyset$; $\partial^{-}_{\blacksquare} \leftarrow \emptyset$
14: **for** $\square l \in HB$ **do**
15: **if** $R^{\square}[l] \cup R^{\mathsf{B},\square}[l] = \emptyset$ **then** REFUTED(l, $\square$)
16: **end for**
17: **for** $r \in R^{\square} \cup R^{\mathsf{B},\square}$ **do**
18: **if** $A(r) = \emptyset$ **then**
19: $r_{inf} \leftarrow \{r \in R : (r, s) \in >, s \in R\}$; $r_{sup} \leftarrow \{s \in R : (s, r) \in >\}$
20: $R_{infd} \leftarrow R_{infd} \cup r_{inf}$
21: Let l be the first literal of $C(r)$ in HB
22: **if** $r_{sup} = \emptyset$ **then**
23: **if** $\square = \mathsf{D}$ **then**
24: PROVED(m, D)
25: **else**
26: REFUTED($\sim l$, $\square$)
27: REFUTED($\sim l$, $\diamond$) for $\diamond$ s.t. Conflict($\square, \diamond$)
28: **if** $R^{\square}[\sim l] \cup R^{\mathsf{B},\square}[\sim l] \cup R^{\blacksquare}[\sim l] \setminus R_{infd} \subseteq r_{inf}$, for $\blacksquare$ s.t. Conflict($\blacksquare, \square$) **then**
29: PROVED(m, $\square$)
30: **end if**
31: **end if**
32: **end if**
33: **end if**
34: **end for**
35: $\partial^{+}_{\blacksquare} \leftarrow \partial^{+}_{\blacksquare} \setminus +\partial_{\blacksquare}$; $\partial^{-}_{\blacksquare} \leftarrow \partial^{-}_{\blacksquare} \setminus -\partial_{\blacksquare}$
36: $+\partial_{\blacksquare} \leftarrow +\partial_{\blacksquare} \cup \partial^{+}_{\blacksquare}$; $-\partial_{\blacksquare} \leftarrow -\partial_{\blacksquare} \cup \partial^{-}_{\blacksquare}$
37: **until** $\partial^{+}_{\blacksquare} = \emptyset$ and $\partial^{-}_{\blacksquare} = \emptyset$
38: **return** $(+\partial_{\blacksquare}, -\partial_{\blacksquare})$

and elimination to obtain a simplified but equivalent theory at each step. Proving a literal reveals which rules should be discarded or reduced in head or body.

A rule can "fire" (line 18) when the antecedent (body) of the rule has been truncated to length zero from previous iterations of the algorithm. If literal q is proved then at the next step: (1) then A(r) does not depend on l ∈ A(r) anymore, and it can be removed. (2) Any rule in which the intersection with the complement set of the proved literal that is not empty is discarded. Superiority tuples are now useless because q was proved true (3) if q = Om, chains for obligation rules can be truncated at m.

Lines 1-5 set up data structures, lines 6-9 handle facts as proved literals. The algorithm checks whether there are rules with an empty body: those are applicable and produce conclusions in a mode X. It also checks for the complement of the same mode, if so then those are weaker than the applicable ones. When a literal is evaluated to provable the algorithm calls PROVED and when a literal is rejected, REFUTED is

called. The two subroutines reduce the complexity of the theory.

Algorithm 2 PROVED

```
1: procedure PROVED(l ∈ Lit, □ ∈ MOD)
2:     ∂+□ ← ∂+□ ∪ {l}; l■ ← l■ ∪ {+□}
3:     HB ← HB \ {□l}
4:     if □ ≠ D then REFUTED(∼l, □)
5:     if □ = B then REFUTED(∼l, I)
6:     if □ ∈ {B, O} then REFUTED(∼l, SI)
7:     R ← {r : A(r) \ {□l, ¬□∼l} ↪ C(r)| r ∈ R, A(r) ∩ □̃l = ∅}
8:     R^{B,□} ← {r : A(r) \ {l} ↪ C(r)|r ∈ R^{B,□}, ∼l ∉ A(r)}
9:     ><-> \{(r, s), (s, r) ∈> | A(r) ∩ □̃l ≠ ∅}
10:    switch (□)
11:        case B:
12:            R^X ← {A(r) ⇒_X C(r)!l| r ∈ R^X[l, n]} with X ∈ {O, I}
13:            if +O ∈ ∼l■ then R^O ← {A(r) ⇒_O C(r) ⊖ ∼l| r ∈ R^O[∼l, n]}
14:            if −O ∈ ∼l■ then R^SI ← {A(r) ⇒_SI C(r)!l| r ∈ R^SI[l, n]}
15:        case O:
16:            R^O ← {A(r) ⇒_O C(r)!∼l ⊖ ∼l| r ∈ R^O[∼l, n]}
17:            if −B ∈ l■ then R^O ← {A(r) ⇒_O C(r) ⊖ l| r ∈ R^O[l, n]}
18:            if −B ∈ ∼l■ then R^SI ← {A(r) ⇒_SI C(r)!l| r ∈ R^SI[l, n]}
19:        case D:
20:            if +D ∈ ∼l■ then
21:                R^G ← {A(r) ⇒_G C(r)!l ⊖ l| r ∈ R^G[l, n]}
22:                R^G ← {A(r) ⇒_G C(r)!∼l ⊖ ∼l| r ∈ R^G[∼l, n]}
23:            end if
24:        otherwise:
25:            R^□ ← {A(r) ⇒_□ C(r)!l| r ∈ R^□[l, n]}
26:            R^□ ← {A(r) ⇒_□ C(r) ⊖ ∼l| r ∈ R^□[∼l, n]}
27:    end switch
28: end procedure
```

Algorithm 3 REFUTED

```
1: procedure REFUTED(l ∈ Lit, □ ∈ MOD)
2:     ∂−□ ← ∂−□ ∪ {l}; l■ ← l■ ∪ {−□}
3:     HB ← HB \ {□l}
4:     R ← {r : A(r) \ {¬□l} ↪ C(r)| r ∈ R, □l ∉ A(r)}
5:     R^{B,□} ← R^{B,□} \ {r ∈ R^{B,□} : l ∈ A(r)}
6:     ><-> \{(r, s), (s, r) ∈> |□l ∈ A(r)}
7:     switch (□)
8:         case B:
9:             R^I ← {A(r) ⇒_I C(r)!∼l|r ∈ R^I[∼l, n]}
10:            if +O ∈ l■ then R^O ← {A(r) ⇒_O C(r) ⊖ l|r ∈ R^O[l, n]}
11:            if −O ∈ l■ then R^SI ← {A(r) ⇒_SI C(r)!∼l|r ∈ R^SI[∼l, n]}
12:        case O:
13:            R^O ← {A(r) ⇒_O C(r)!l ⊖ l|r ∈ R^O[l, n]}
14:            if −B ∈ l■ then R^SI ← {A(r) ⇒_SI C(r)!∼l|r ∈ R^SI[∼l, n]}
15:        case D:
16:            R^X ← {A(r) ⇒_X C(r) ⊖ l|r ∈ R^X[l, n]} with X ∈ {D, G}
17:        otherwise:
18:            R^□ ← {A(r) ⇒_□ C(r) ⊖ l|r ∈ R^□[l, n]}
19:    end switch
20: end procedure
```

4 More Intuitive Explanation of defeasible extension algorithm

This section aims to outline the Defeasible Extension algorithm in words. Can also be found as comments in the python code.

A Defeasible Extension Algorithm class consists of **Rules**, a map, where the key is the modality and value is a list of corresponding rules of the key modality), **R** for rules of modality that are of goal-like attitudes, and produce outcomes. For example, Obligation O drives an agent towards a goal by eliminating violations of O and thus is of a goal-like attitude. **LiteralMods**, a map where key is the modality and value is a list of literals that ontake this modality. For example, given (O,visit), O is modality and visit is the literal. **conclusions**, both global and local defeasible conclusions. The local defeasible conclusions are updated at each iteration and are used to update the global sets.

For every outcome rule, the algorithm makes a copy of the same rule corresponding to a goal like attitude. Since beliefs can be obligations as well, for example, rules with belief as modality are copied to ontake the modality of the conversion to. Since beliefs are also facts, or how the agent see the world, we immediately mark belief rules as Proved, or Refuted. Keeping in mind the closed world assumption.

Then, for every literal l in the Herbrand Base, it is necessary that the literal l is reachable from some belief rule. If it is not reachable, then the literal cannot be derived from what the agent thinks is achievable, and thus the literal l is marked as Refuted. Rules that can "fire" have an empty body, or antecedent. This means that all antecedents in the head were either proved and removed, or refuted, and removed. In any case, all "if" conditions are satisfied and the "then" consequents can fire. If a rule r is proven because the body is empty, then we can check if there is any rule s that is superior to r. If there is no such s then the complement of r, $\bar{r}$, is Refuted. If r has modality Desire then it is Proved. The complement is refuted in modality of the rule, since logical agents do not exhibit wishful thinking, or wanting the impossible, apart from Desire modalities. We also check for conflicts with the rule modality for which the literal l in

the rule consequents is observed. All literals of rules with conflicting modality are refuted *l* since we know there is no superior rule to *r*.

5 Implementation Details

Apart from the algorithm class Defeasible Extension, there are two classes: **Rule** and **Literal**. A rule has a *modality* in belief, obligation, desire, goal, intention, outcome, social intention, *antecedent* list of literal premises, *consequent* chain of literal outcomes, *superiors* set of rules that dominate current rule, and *inferiors* set of rules that this current rule dominates. Literals make up a rule's antecedents and consequents. A literal has a *prop*, which is either '+' or '-', *mod* which corresponds to the rule it belongs to, *lit* which is the literal. For example: literal ('-',D,visit_john) means the agent does not have the desire to visit John.

6 Links

Original paper: https://arxiv.org/abs/1512.04021/
Github: https://github.com/hhyao00/defeasible-extension

References

[1] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming: An argumentative approach. *Theory Pract. Log. Program.*, 4(2):95–138, January 2004.

[2] Guido Governatori, Francesco Olivieri, Simone Scannapieco, Antonino Rotolo, and Matteo Cristani. The rationale behind the concept of goal. *TPLP*, 16:296–324, 2016.

[3] Robert Koons. Defeasible reasoning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition, 2017.