$\mathcal{ALE}$ Defeasible Description Logic

Pakornpong Pothipruk[1] and Guido Governatori[2]

School of ITEE, University of Queensland, Australia
[1]pkp@itee.uq.edu.au
[2]guido@itee.uq.edu.au

Abstract. One of Semantic Web strengths is the ability to address incomplete knowledge. However, at present, it cannot handle incomplete knowledge directly. Also, it cannot handle non-monotonic reasoning. In this paper, we extend $\mathcal{ALC}^-$ Defeasible Description Logic with existential quantifier, i.e., $\mathcal{ALE}$ Defeasible Description Logic. Also, we modify some parts of the logic, resulting in an increasing efficiency in its reasoning.

1 Introduction

Description logics, which is a language based on first order logic, cannot handle incomplete and inconsistent knowledge well. In fact, it is often that complete and consistent information or knowledge are very hard to obtain. Consequently, there is an urgent need to extend description logics with a non-monotonic part which can handle incomplete and inconsistent knowledge better than themselves. Defeasible logic, developed by Nute [6], is a well-established nonmonotonic reasoning system. Its outstanding properties are low computational complexity [4,5] and ease of implementation. Thus, this logic is the nonmonotonic part of our choice. Governatori [3] proposes a formalism to extend $\mathcal{ALC}^-$ with defeasibility. He combines the description logic formalism with the defeasible logic formalism, creating a logic that can handle both monotonic and non-monotonic information coherently. We extend this work to be able to handle existential quantification.

2 Introduction to Defeasible Logic

Defeasible logic handles non-monotonicity via five types of knowledge: Facts, Strict Rules (monotonic rules, e.g., $LOGISTIC\_MANAGER(x) \rightarrow EMPLOYEE(x)$), Defeasible Rules (rules that can be defeated, e.g., $MANAGE\_DELIVER(x) \Rightarrow EMPLOYEE(x)$), Defeaters (preventing conclusions from defeasible rules, e.g., $TRUCK\_DRIVER(x) \leadsto \neg EMPLOYEE(x)$), and a Superiority Relation (defining priorities among rules). Note that We consider only propositional rules. Rule with free variables will be propositionalized, i.e., it is interpreted as the set of its grounded instances. Due to space limitation, see [6,2,1] for detailed explanation of defeasible logic.

Defeasible logic is proved to have well behavior: 1) coherence and 2) consistence [3,2]. Furthermore, the consequences of a propositional defeasible theory D can be derived in $O(N)$ time, where N is the number of propositional literals in D [4]. In fact,

A. Sattar and B.H. Kang (eds)
19th Australasian Joint Conference on Artificial Intelligence, LNAI 4304, pp. 110–119., pp. 110–119, 2006.

the linear algorithm for derivation of a consequence from a defeasible logic knowledge base exploits a transformation algorithm, presented in [1], to transform an arbitrary defeasible theory D into a *basic defeasible theory* D_b.Given a defeasible theory D, the tranformation procedure $Basic(D)$ consists of three steps: 1) normalize the defeasible theory D, 2) eliminate defeaters: simulating them using strict rules and defeasible rules, and 3) eliminate the superiority relation: simulating it using strict rules and defeasible rules. The procedure then returns a transformed defeasible theory D_b. Given a basic defeasible theory D_b, defeasible proof conditions in [3] are simplified as follows:

$+\partial$**:** If $P(i+1) = +\partial q$ then either

1. $+\Delta q \in P(1..i)$ or
2. (a) $\exists r \in R[q], \forall a \in A(r) : +\partial a \in P(1..i)$ and
   (b) $\forall s \in R[\sim q], \exists a \in A(s) : -\partial a \in P(1..i)$

$-\partial$**:** If $P(i+1) = -\partial q$ then

1. $-\Delta q \in P(1..i)$ and either
2. (a) $\forall r \in R_{sd}[q], \exists a \in A(r) : -\partial a \in P(1..i)$ or
   (b) $\exists s \in R[\sim q], \forall a \in A(s) : +\partial a \in P(1..i)$

3 $\mathcal{ALC}^-$ Defeasible Description Logic

In this section, we introduce an $\mathcal{ALC}^-$ defeasible description logic [3], i.e., an extension of $\mathcal{ALC}^-$ description logic with defeasible logic. Also, we introduce several adjustments for the logic.

Like $\mathcal{ALC}$ description logic, $\mathcal{ALC}^-$ knowledge base consists of a Tbox and Abox(es), i.e., $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$. The Abox $\mathcal{A}$, containing a finite set of concept membership assertions ($a : C$ or $C(a)$) and role membership assertions ($(a,b) : R$ or $R(a,b)$), corresponds to the set of facts F in defeasible logic. The Tbox $\mathcal{T}$ contains a finite set of axioms of the form: $C \sqsubseteq D \mid C \doteq D$, where C and D are concept expressions. Concept expressions are of the form: $A \mid \top \mid \perp \mid \neg(atomiconly)C \mid C \sqcap D \mid \forall R.C$ where A is an atomic concept or concept name, C and D are concept expressions, R is a simple role name, $\top$ (top or full domain) represents the most general concept, and $\perp$ (bottom or empty set) represents the least general concept. Their semantics are similar to ones in $\mathcal{ALC}$ description logic. In fact, the Tbox $\mathcal{T}$ corresponds to the set of strict rules in defeasible logic. Governatori [3] shows how can we transform $\mathcal{ALC}^-$ Tbox to a set of strict rules, using a set of axiom pairs and a procedure *ExtractRules*. Basically, an inclusion axiom: $C_1 \sqcap C_2 \sqcap \ldots \sqcap C_m \sqsubseteq D_1 \sqcap D_2 \sqcap \ldots \sqcap D_n$ in the Tbox $\mathcal{T}$ is transformed into an axiom pair (ap): $\langle \{C_1(x), C_2(x), \ldots, C_m(x)\}, \{D_1(x), D_2(x), \ldots, D_n(x)\} \rangle$. Eventually, for the Tbox $\mathcal{T}$, we get the set AP of all axiom pairs. Then, the procedure *ExtractRules* in Algorithm 1 is used to transform AP into a set of strict rules in defeasible logic.

Algorithm 1 *ExtractRules*(AP):

if $ap = \langle\{C(x)\},\{D(x)\}\rangle \in AP$ **then**
  $C(x) \rightarrow D(x) \in R_s$
  $\neg D(x) \rightarrow \neg C(x) \in R_s$
  $AP = AP - \{ap\}$
else if $ap = \langle\{C_1(x_{C1}),\ldots,C_{Cn}(x_n),R_1(x_{R1},y_{R1}),\ldots,$
$R_m(x_{Rm},y_{Rm})\},\{D_1(x_{D1}),\ldots,D_k(x_{Dk}),\forall S_1.D_{S1}(x_{S1})$
$\ldots,\forall S_k.D_{Sk}(x_{Sk})\}\rangle \in AP$ **then**
  for all $I \in \{D_1,\ldots,D_k\}$ **do**
    $C_1(x_{C1}),\ldots,C_n(x_{Cn}),R_1(x_{R1},y_{R1}),\ldots,R_m(x_{Rm},y_{Rm})$
    $\rightarrow I(x_I) \in R_s$
  end for
  for $1 \leq i \leq k$ **do**
    $AP = AP - \{ap\} \cup$
    $$\left\{\begin{array}{l}\langle\{C_1(x_{C1}),\ldots,C_n(x_{Cn}),R_1(x_{R1},y_{R1}),\\ \ldots,R_m(x_{Rm},y_{Rm}),S_i(x_{Si},y_{Si})\},\{D_1^{Si}(y_{Si}),\\ \ldots,D_l^{Si}(y_{Si}),\forall T_1^{Si}.D_1^{Si}(y_{Si}),\ldots,\forall T_p^{Si}.D_p^{Si}(y_{Si})\}\rangle\end{array}\right\},$$
    where
      y_{Si} is a new variable (not in ap), and
      $D_i = D_1^{Si} \sqcap \ldots \sqcap D_l^{Si} \sqcap \forall T_1^{Si}.D_1^{Si} \sqcap \ldots \sqcap \forall T_p^{Si}.D_p^{Si}$
  end for
end if
if AP is not an empty set **then**
  ExtractRules(AP)
end if
return R_s

This procedure progresses recursively until all axiom pairs in AP are transformed. Here is an example: let $\mathcal{T} = \{C_1 \sqcap \forall R_1.C_2 \sqsubseteq C_3 \sqcap \forall R_2.(C_4 \sqcap \forall R_3.C_5)\}$. Here are the steps for strict rules generation.

Step 1: $AP = \{\langle\{C_1(x),\forall R_1.C_2(x)\},\{C_3(x),\forall R_2.(C_4 \sqcap \forall R_3.C_5)(x)\}\rangle\}$.

Step 2: $ExtractRules(AP) : C_1(x),\forall R_1.C_2(x) \rightarrow C_3(x) \in R_s$.

Step 3: $AP = \{\langle\{C_1(x),\forall R_1.C_2(x),R_2(x,y)\},$
$\{C_4(y),\forall R_3.C_5(y)\}\rangle\}$.

Step 4: $ExtractRules(AP) : C_1(x),\forall R_1.C_2(x),R_2(x,y) \rightarrow C_4(y) \in R_s$.

Step 5: $AP = \{\langle\{C_1(x),\forall R_1.C_2(x),R_2(x,y),R_3(y,z)\}$
$,\{C_5(z)\}\rangle\}$.

Step 6: $ExtractRules(AP) : C_1(x),\forall R_1.C_2(x),R_2(x,y),$
$R_3(y,z) \rightarrow C_5(z) \in R_s$.

Step 7: $AP = \{\}$, the procedure ends here.

Notice that the *ExtractRules* procedure takes care of description logic's universal quantified concepts, occurring on the RHS of the Tbox axioms, by transforming those axioms into first order logic rules. However, description logic's universal quantified concepts occurring on the LHS of the Tbox axioms still remain in those rules. At this point, we need additional inference rules to deal with the remaining universal quantified concepts. However, universal quantified concepts' semantics take into account all individuals in the description logic knowledge base, in particular the Abox. Consequently, the proof conditions for universal quantified concepts will incorporate the domain of Abox $\mathcal{A}$, i.e., $\Delta^{\mathcal{I}}_{\mathcal{A}}$ (see [7]), in themselves as follows:

$+\Delta\forall R.C$**:** If $P(i+1) = +\Delta\forall R.C(a)$ then $\forall b \in \Delta^{\mathcal{I}}_{\mathcal{A}}$,
1)$-\Delta R(a,b)$ or 2) $+\Delta C(b)$.
$-\Delta\forall R.C$**:** If $P(i+1) = -\Delta\forall R.C(a)$ then $\exists b \in \Delta^{\mathcal{I}}_{\mathcal{A}}$,
1) $+\Delta R(a,b)$ and 2) $-\Delta C(b)$.
$+\partial\forall R.C$**:** If $P(i+1) = +\partial\forall R.C(a)$ then $\forall b \in \Delta^{\mathcal{I}}_{\mathcal{A}}$,
1) $-\partial R(a,b)$ or 2) $+\partial C(b)$.
$-\partial\forall R.C$**:** If $P(i+1) = -\partial\forall R.C(a)$ then $\exists b \in \Delta^{\mathcal{I}}_{\mathcal{A}}$,
1) $+\partial R(a,b)$ and 2) $-\partial C(b)$.

Now, we have a complete formalism to deal with $\mathcal{ALC}^-$ literals. Since a rule, however, can consists literal(s) with variable(s), there can be *vague rules* such as: $C(x), D(y) \rightarrow E(z)$. In this rule, variables in one literal are not bounded/connected with variables in any other literals. Given a rule, we say that the rule is a *variable connected rule* if, for every literal in the rule, there exists another literal which has the same variables as the literal, and a *vague rule* otherwise . Here are examples of variable connected rules: $C(x), D(x) \rightarrow E(x)$ and $C(x), R(x,y) \Rightarrow P(y,z)$. In this thesis, we only allow variable connected rules in the knowledge base.

Governatori [3] also shows that $\mathcal{ALC}^-$ defeasible description logic is coherent and consistent. He also proves that the complexity of $\mathcal{ALC}^-$ defeasible description logic w.r.t. a defeasible description theory D is $O(n^4)$, where n is the number of symbols in D. This proof is based on two steps: 1) propositionalization the theory, and 2) analyze proof conditions for universal quantified concepts. For the first step, in short, since the logic allows roles (binary predicates), the size of resulting theory is $O((\Delta^{\mathcal{I}}_{\mathcal{A}})^2)$, assuming the number of rules is much less than the size of $\Delta^{\mathcal{I}}_{\mathcal{A}}$. For the second step, the proof conditions for universal quantified concepts are embedded in the propositionalization procedure. Let n be the size of $\Delta^{\mathcal{I}}_{\mathcal{A}}$. For each universal quantified literal $\forall R.C(x)$ in an antecedent of a rule, i.e., U, create the following propositional auxiliary rules:

for every a_i in $\Delta^{\mathcal{I}}_{\mathcal{A}}$,
$RC(a_i,a_1),\ldots,RC(a_i,a_n) \rightarrow \forall R.C(a_i)$
for every a_j in $\Delta^{\mathcal{I}}_{\mathcal{A}}$,
$\sim R(a_i,a_j) \rightarrow RC(a_i,a_j)$
$C(a_j) \rightarrow RC(a_i,a_j)$.

,using the $\mathit{AuxiliaryRules}(U)$ procedure. In fact, each universal quantified literal in antecedent adds at most $(\Delta^{\mathcal{I}}_{\mathcal{A}})^2$ auxiliary rules to the theory. Thus, the size of the new propositionalized theory is $O((\Delta^{\mathcal{I}}_{\mathcal{A}})^4)$ in size of the original theory D. As mentioned before that a consequence of a propositional defeasible theory D can be derived in $O(N)$ time, where N is the number of propositional literals in D [4], thus, a consequence of a propositional $\mathcal{ALC}^-$ defeasible description logic theory can be derived in polynomial time, i.e., $O(N(\Delta^{\mathcal{I}}_{\mathcal{A}})^4)$ time.

In summary, to reason with an $\mathcal{ALC}^-$ defeasible description logic, we must do the following steps:

Step 1: Transform $\mathcal{T}$ to R_s using the *ExtractRules* procedure, and transform $\mathcal{A}$ to F, hence, $D_{DDL} = \langle \mathcal{T}, \mathcal{A}, R, > \rangle$ is reduced to $D = \langle F, R, > \rangle$. Given a theory D_{DDL} and a conclusion T (i.e., a query), this step is designed such that $D_{DDL} \vdash T$ iff $D \vdash T$ [3].

Step 2: Propositionalize D to D_{prop}, including propositional auxiliary rules for universal quantified literals ($\forall R.C(x)$) in antecedents of rules. Given a theory D and a conclusion T (i.e., a query), this step guarantees that $D \vdash T$ iff $D_{prop} \vdash T$. It is easy to verify that this statement follows immediately from the first step statement plus the nature of propositionalization.

Step 3: Apply $LinearProve(Basic(D_{prop}))$, the linear algorithm. Note that $\mathcal{A}$ and $\mathcal{T}$ are normalized in this step.

Even the complexity of the logic is polynomial in time, in practice, the reasoning process on $\mathcal{ALC}^-$ defeasible description logic still suffers a huge number of additional propositionalized rules, regardless of the linear time complexity reasoning for a propositionalized theory. Consequently, it is essential to optimize the reasoning, especially the propositionalization step (Step 2).

Before we proceed with the optimization, we show that the propositionalization step is not trivial. First, we re-consider (variable-connected) simple strict rules:

$$\begin{aligned} r1 &: C(x), D(x) && \rightarrow E(x) \\ r2 &: C(x), R(x,y) && \rightarrow E(x) \\ r3 &: C(x), D(x) && \rightarrow R(x,y) \\ r4 &: R(x,y), P(y,z) && \rightarrow C(x) \end{aligned}$$

As you can notice that there is no variable quantifier in the rules. In fact, the above four rules are equal to the following rules:

$$\begin{aligned} r1 &: \forall x && C(x), D(x) && \rightarrow E(x) \\ r2 &: \forall x \forall y && C(x), R(x,y) && \rightarrow E(x) \\ r3 &: \forall x \forall y && C(x), D(x) && \rightarrow R(x,y) \\ r4 &: \forall x \forall y \forall z && R(x,y), P(y,z) && \rightarrow C(x) \end{aligned}$$

Let n be the number of individuals in the domain $\Delta^{\mathcal{I}}_{\mathcal{A}}$. In rule $r1$, there is only one variable, thus it is propositionalized to n propositional rules. In rule $r2$ and $r3$, there are two variables in each rule, thus each rule is propositionalized to n^2 propositional rules. In rule $r4$ there are three variables. In fact, the rule $r4$ is equivalent to $\forall x \forall y \forall z RP(x,y,z) \rightarrow C(x)$, where $\forall x \forall y \forall z RP(x,y,z) \equiv \forall x \forall y \forall z R(x,y) \wedge P(y,z)$. Consequently, the rule $r4$ is propositionalized to n^3 propositional rules, which will make the complexity of a conclusion derivation in $\mathcal{ALC}^-$ defeasible description logic increase to $O(n^5)$. In fact, the size of propositionalized theory is $O(n^{n_v})$ of the size of the original theory, where n_v is the maximum number of variables in rules.

Second, we re-consider (variable-connected) strict rules with universal quantified concepts (e.g., $\forall R.C(x)$) in their consequents. For example, the rule $r1 : \forall x C(x) \rightarrow \forall R.\forall P.D(x)$ is transformed to $r1 : \forall x \forall y \forall z C(x), R(x,y), P(y,z) \rightarrow D(z)$, using the *ExtractRule* procedure, which is, in turn, propositionalized to n^3 propositional rules. In fact, given a rule of this kind, propositionalization will generate additional $n^{d_\forall + 1}$ rules, where $d_\forall$ is depth of the nested universal quantified consequent, e.g., $d_\forall = 2$ for $\forall R.\forall P.D(x)$.

Third, we re-consider (variable-connected) strict rules with universal quantified concepts (e.g., $\forall R.C(x)$) in their antecedents. For example, the rule $r1 : \forall x \forall R.\forall P.D(x) \rightarrow C(x)$ is propositionalized to n propositional rules, plus the following propositional auxiliary rules:

$$
\begin{array}{l}
\text{for every } a_i \text{ in } \Delta^{\mathcal{I}}_{\mathcal{A}}, \\
\quad R\forall P.C(a_i,a_1),\ldots,R\forall P.C(a_i,a_n) \rightarrow \forall R.\forall P.C(a_i) \\
\quad \text{for every } a_j \text{ in } \Delta^{\mathcal{I}}_{\mathcal{A}}, \\
\quad\quad \sim R(a_i,a_j) \rightarrow R\forall P.C(a_i,a_j) \\
\quad\quad \forall P.C(a_j) \rightarrow RC(a_i,a_j) \\
\quad\quad PC(a_j,a_k),\ldots,PC(a_j,a_k) \rightarrow \forall P.C(a_j) \\
\quad\quad \text{for every } a_k \text{ in } \Delta^{\mathcal{I}}_{\mathcal{A}}, \\
\quad\quad\quad \sim P(a_j,a_k) \rightarrow PC(a_j,a_k) \\
\quad\quad\quad C(a_k) \rightarrow PC(a_j,a_k).
\end{array}
$$

It is easy to verify that the number of propositional auxiliary rules generated is $n(2n+1)^2$, or about n^3. In fact, given a rule of this kind, propositionalization will generate additional $n(2n+1)^{d_\forall}$ auxiliary rules for each universal quantified antecedent, where $d_\forall$ is depth of the nested universal quantified antecedent, e.g., $d_\forall = 2$ for $\forall R.\forall P.D(x)$.

At this point, the size of the new propositionalized theory D_{prop} can exceed $O((\Delta^{\mathcal{I}}_{\mathcal{A}})^4)$ in size of the original theory D. In fact, the size of the new propositionalized theory D_{prop} is $O((\Delta^{\mathcal{I}}_{\mathcal{A}})^{n_{max}})$ in size of the original theory D, where n_{max} is the maximum of $(n_v + d^{RHS}_\forall, d^{LHS}_\forall + 3)$, n_v is the maximum number of variables in each rule, $d^{LHS}_\forall$ is depth of the biggest nested universal quantified antecedent, and $d^{RHS}_\forall$ is depth of the biggest nested universal quantified consequent. However, we can modify the *ExtractRules* and *AuxiliaryRules* procedures such that size of the resulting propositionalized theory is $O(n^3)$ and $O(n^4)$ the size of the original theory respectively.

Since we allow only variable-connected rules with at most two variables, the *ExtractRules* procedure is transformed to $ExtractRules_2$. In the algorithm 2, we introduce the notion of *intermediate literals*. The intermediate literal correctness follows immediately from the semantics of universal quantified concept in description logic. Here is a simple example demonstrating a usage of the algorithm 2: let $\mathcal{T} = \{C_1 \sqcap \forall R_1.C_2 \sqsubseteq C_3 \sqcap \forall R_2.(C_4 \sqcap \forall R_3.C_5)\}$. Here are the steps for strict rules generation.

Step 1: $AP = \{\langle\{C_1(x), \forall R_1.C_2(x)\}, \{C_3(x), \forall R_2.(C_4 \sqcap \forall R_3.C_5)(x)\}\rangle\}$.

Step 2: $ExtractRules(AP) : \forall x, C_1(x), \forall R_1.C_2(x) \rightarrow C_3(x) \in R_s$.

Step 3: $AP = \{\langle\{C_1(x), \forall R_1.C_2(x), R_2(x,y)\}, \{C_4(y)$
$, R_3C_5(y)\}\rangle, \langle\{R_3C_5(x)\}, \{\forall R_3.C_5(x)\}\rangle\}$.

Step 4: $ExtractRules(AP)$:

- $\forall x \forall y, C_1(x), \forall R_1.C_2(x), R_2(x,y) \rightarrow C_4(y) \in R_s$.
- $\forall x \forall y, C_1(x), \forall R_1.C_2(x), R_2(x,y) \rightarrow R_3C_5(y) \in R_s$.

Step 5: $AP = \{\langle\{R_3C_5(x)\}, \{\forall R_3.C_5(x)\}\rangle\}$.

Step 6: $ExtractRules(AP)$: get no rule.

Step 7: $AP = \{\langle\{R_3C_5(x), R_3(x,y)\}, \{C_5(y)\}\rangle\}$

Step 8: $ExtractRules(AP)$: $\forall x \forall y, R_3C_5(x), R_3(x,y) \rightarrow C_5(y) \in R_s$

Step 9: $AP = \{\}$, the procedure ends here.

Algorithm 2 $ExtractRules_2(AP)$:

if $ap = \langle\{C(x)\},\{D(x)\}\rangle \in AP$ **then**
  $\forall x, C(x) \rightarrow D(x) \in R_s$
  $\forall x, \neg D(x) \rightarrow \neg C(x) \in R_s$
  $AP = AP - \{ap\}$
else if $ap = \langle\{C_1(x),\ldots,C_{Cn}(x),R_1(x,y),\ldots,R_m(x,y)\},$ $\{D_1(x),\ldots,D_k(x)\}\rangle \in AP$ **then**
  for all $I \in \{D_1,\ldots,D_k\}$ **do**
    $\forall x \forall y, C_1(x),\ldots,C_{Cn}(x),R_1(x,y),\ldots,R_m(x,y) \rightarrow I(x)$
  end for
  $AP = AP - \{ap\}$
else if $ap = \langle\{C_1(x),\ldots,C_{Cn}(x),R_1(x,y),\ldots,R_m(x,y)\},$ $\{D_1(x),\ldots,D_k(x),\forall S_1.D_{S1}(x),\ldots,\forall S_k.D_{Sk}(x)\}\rangle \in AP$ **then**
  for all $I \in \{D_1,\ldots,D_k\}$ **do**
    $\forall x \forall y, C_1(x),\ldots,C_n(x),R_1(x,y),\ldots,R_m(x,y)$ $\rightarrow I(x) \in R_s$
  end for
  for $1 \leq i \leq k$ **do**
    $AP = AP - \{ap\} \cup$
    $$\left\{\begin{array}{l} \langle\{C_1(x),\ldots,C_n(x),R_1(x,y),\ldots,R_m(x,y), \\ S_i(x,z)\},\{D_1^{Si}(z),\ldots,D_l^{Si}(z),T_1^{Si}D_1^{Si}(z),\ldots, \\ T_p^{Si}D_p^{Si}(z)\}\rangle\langle\{T_1^{Si}D_1^{Si}(x)\}, \\ \{\forall T_1^{Si}D_1^{Si}(x)\}\rangle\ldots\langle\{T_p^{Si}D_p^{Si}(x)\}, \\ \{\forall T_p^{Si}D_p^{Si}(x)\}\rangle \end{array}\right\},$$
    where
      y and z are new variables (not in ap),
      $D_i = D_1^{Si} \sqcap \ldots \sqcap D_l^{Si} \sqcap \forall T_1^{Si}.D_1^{Si} \sqcap \ldots \sqcap \forall T_p^{Si}.D_p^{Si}$,
      and $T_1^{Si}D_1^{Si}(x),\ldots,T_p^{Si}D_p^{Si}(x)$ are *intermediate literals*.
  end for
end if
if AP is not an empty set **then**
  $ExtractRules_2(AP)$
end if
return R_s

It is easy to verify that the $ExtractRules_2$ procedure in algorithm 2 generates the resulting propositionalized theory of the size $O(n^3)$ the size of the original theory.

Regarding the *AuxiliaryRules* procedure, since we allow only variable-connected rules with at most two variables, for each universal quantified literal $U = \forall R.C(x)$ in an antecedent of a rule, call the procedure $AuxiliaryRules_2(U)$:

Algorithm 3 *AuxiliaryRules*$_2(U)$:

$\forall x, RC(x,a_1),...,RC(x,a_n) -> \forall R.C(x) \in R_s$
$\forall x \forall y, \sim R(x,y) -> RC(x,y) \in R_s$
$\forall x \forall y, C(y) -> RC(x,y) \in R_s$.
if $C(y)$ is a universal quantified literal, then
  AuxiliaryRules$_2(C(x))$

, where $\Delta_{\mathcal{A}}^{\mathcal{I}}$ is $\{a_1,...,a_n\}$, and n is the size of $\Delta_{\mathcal{A}}^{\mathcal{I}}$. After this, we can propositionalize the additional auxiliary rules as usual. It is easy to verify that the size of of the propositionalized theory is $O(n^4)$ the size of the original theory.

In summary, if we limit the number of variable $n_v = 2$, and use the above specifications for universal quantified literals (both antecedent and consequent), we will regain complexity of the modified $\mathcal{ALC}^-$ defeasible description logic to be $O(n^4)$ again.

Fourth, we show how a defeater can be propositionalized. A defeater is only used to prevent a conclusion, thus its consequent is a negative literal. In $\mathcal{ALC}^-$ defeasible description logic, only atomic negation is allowed. In addition, we only allow variable-connected rules with at most two free variables in each rule. Consequently, a defeater can be of the form $\forall x(\forall y), LHS \rightsquigarrow \neg C(x) \mid \forall x \forall y, LHS \rightsquigarrow \neg R(x,y)$. In propositionalization, a defeater is treated as a rule. For example, let $\Delta_{\mathcal{A}}^{\mathcal{I}} = \{a,b,c\}$, a defeater $\forall x, C(x) \rightsquigarrow \neg D(x)$ is propositionalized to: $C(a) \rightsquigarrow \neg D(a) \in R_{dft}, C(b) \rightsquigarrow \neg D(b) \in R_{dft}, C(c) \rightsquigarrow \neg D(c) \in R_{dft}$. A propositionalized defeater will prevent conclusion by not firing defeasible rule(s) whose head(s) has(have) literal(s) which is(are) negativity of the defeater head if all literals in the defeater body are provable. Since, a defeater is propositionalized in the same way as a rule is, it will prevent correct propositionalized defeasible rule(s) from firing. Hence, the size of propositionalized R_{dft} is $O(n^2)$ the size of original R_{dft}. Thus, complexity of the $\mathcal{ALC}^-$ defeasible description logic is still $O(n^4)$.

Lastly, we show how can we extend the superiority relation to cover the propositionalized rules. Since the superiority relation is defined over pairs of rules, in particular defeasible rules, which have contradictory heads, we only need to extend the superiority relation to cover the corresponding pairs of propositionalized rules. We illustrate this fact by a simple example. Let $\Delta_{\mathcal{A}}^{\mathcal{I}} = \{a,b,c\}$, we have a set of rules: $r1 : \forall x, C(x) \Rightarrow E(x), r2 : \forall x, D(x) \Rightarrow \neg E(x)$, and the superiority relation $> = \{\langle r1, r2 \rangle\}$. The set of rules are propositionalized to: $r1a : C(a) \Rightarrow E(a), r1b : C(b) \Rightarrow E(b), r1c : C(c) \Rightarrow E(c), r2a : D(a) \Rightarrow \neg E(a), r2b : D(b) \Rightarrow \neg E(b), r2c : D(c) \Rightarrow \neg E(c)$, and the extended superiority relation $> = \{\langle r1a, r2a \rangle, \langle r1b, r2b \rangle, \langle r1c, r2c \rangle\}$. Hence, the size of extended superiority relation is $O(n^2)$ the size of original superiority relation. Thus, complexity of the $\mathcal{ALC}^-$ defeasible description logic is still $O(n^4)$. Note that the propositionalized R_{dft} and the extended $>$ will be absorbed into R_{sd} in the linear algorithm for a conclusion derivation.

4 $\mathcal{ALE}$ Defeasible Description Logic

In this section, we introduce an $\mathcal{ALE}$ defeasible description logic, i.e., an extension of $\mathcal{ALC}^-$ defeasible description logic with existential quantification constructor.

$\mathcal{ALE}$ knowledge base consists of a Tbox and Abox(es), i.e., $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$. The Abox $\mathcal{A}$, containing a finite set of concept membership assertions ($a : C$ or $C(a)$) and role membership assertions ($(a,b) : R$ or $R(a,b)$), corresponds to the set of facts F in defeasible logic. The Tbox $\mathcal{T}$ contains a finite set of axioms of the form: $C \sqsubseteq D \mid C \doteq D$, where C and D are concept expressions. Concept expressions are of the form: $A \mid \top \mid \bot \mid \neg(atomiconly)C \mid C \sqcap D \mid \forall R.C \mid \exists R.C$ where A is an atomic concept or concept name, C and D are concept expressions, R is a simple role name, $\top$ (top or full domain) represents the most general concept, and $\bot$ (bottom or empty set) represents the least general concept. Their semantics are similar to ones in $\mathcal{ALC}$ description logic. Tbox $\mathcal{T}$ corresponds to the set of strict rules in defeasible logic. The algorithm 4 shows how can we transform $\mathcal{ALE}$ Tbox to a set of strict rules, using a set of axiom pairs and a procedure $ExtractRules_3$. Basically, an inclusion axiom: $C_1 \sqcap C_2 \sqcap \ldots \sqcap C_m \sqsubseteq D_1 \sqcap D_2 \sqcap \ldots \sqcap D_n$ in the Tbox $\mathcal{T}$ is transformed into an axiom pair (ap): $\langle \{C_1(x), C_2(x), \ldots, C_m(x)\}, \{D_1(x), D_2(x), \ldots, D_n(x)\} \rangle$. Eventually, for the Tbox $\mathcal{T}$, we get the set AP of all axiom pairs. Then, the procedure $ExtractRules_3$ is used to transform AP into a set of strict rules in defeasible logic.

Algorithm 4 $ExtractRules_3(AP)$:

if $ap = \langle \{C_1(x), \ldots, C_{Cn}(x), R_1(x,y), \ldots, R_m(x,y)\},$
$\{D_1(x), \ldots, D_k(x), \exists S_1.D_{S1}(x), \ldots, \exists S_k.D_{Sk}(x)\} \rangle \in AP$ **then**
  for all $I \in \{D_1, \ldots, D_k\}$ **do**
    $\forall x \forall y, C_1(x), \ldots, C_n(x), R_1(x,y), \ldots, R_m(x,y)$
    $\rightarrow I(x) \in R_s$
  end for
  $AP = AP - \{ap\} \cup$
  $\left\{ \begin{array}{l} \langle \{C_1(x), \ldots, C_n(x), R_1(x,y), \ldots, R_m(x,y)\}, \\ \{S_1(x,y_1), D_{S1}(y_1), \ldots, S_k(x,y_k), D_{Sk}(y_k)\} \rangle \end{array} \right\},$
else if $ap = \langle \{\exists S_1.D_{S1}(x), \ldots, \exists S_k.D_{Sk}(x)\},$
$\{C_1(x), \ldots, C_n(x)\} \rangle \in AP$ **then**
  $AP = AP - \{ap\} \cup$
  $\left\{ \begin{array}{l} \langle \{S_1 D_{S1}(x), \ldots, S_k D_{Sk}(x)\}, \{C_1(x), \ldots, C_n(x)\} \rangle \\ \langle \{S_1(x,y), D_{S1}(x)\}, \{S_1 D_{S1}(x)\} \rangle, \ldots \\ \langle \{S_k(x,y), D_{Sk}(x)\}, \{S_k D_{Sk}(x)\} \rangle \end{array} \right\},$
else $ExtractRules_2(AP)$
end if
if AP is not an empty set **then**
  $ExtractRules_3(AP)$
end if
return R_s

It is easy to verify that the $ExtractRules_3$ procedure in algorithm 4 generates the resulting propositionalized theory of the size $O(n^4)$ the size of the original theory. Consequently, we get a defeasible description logic with very efficient derivation process for a conclusion.

5 Discussion and Future Works

This paper introduces several modifications to the existing $\mathcal{ALC}^-$ defeasible description logic such that its derivation can be accomplished in $O(n^4)$. This makes the language more useful in practice. Further, we extend the logic with existential quantification constructor, resulting in a new logic, i.e., $\mathcal{ALE}$ defeasible description logic, which can handle more expressive knowledge base in the context of non-monotonic reasoning. Our work is significant because it is a foundation to be extended to higher expressive non-monotonic description logic. In the history of Description Logics, increasing-in-expressiveness description logics have been studied chronologically, in order to find the highest expressive description logic that are still decidable. Also, the maximum bound of tractable logic has been found. However, those are the cases for *monotonic* description logics, not for *nonmonotonic* description logics. This work presents a new result showing a nonmonotonic description logic that is still tractable, i.e., $\mathcal{ALE}$ defeasible description logic. In the near future, we will study how we can add the full negation constructor to the logic, resulting in $\mathcal{ALC}$ defeasible description logic. However, it is still arguable whether nonmonotonic logic can be extended to $\mathcal{SHOIN}(\mathcal{D})$, which is equal to $\mathcal{ALC}^{\mathcal{R}^+}\mathcal{HOIN}(\mathcal{D})$. Transitive roles, role inclusions, one-of operators, inverse roles, qualified number restriction, and concrete domain must be added to the $\mathcal{ALC}$ defeasible description logic, in order to achieve the $\mathcal{SHOIN}(\mathcal{D})$ defeasible description logic. These additions are still open issues.

Acknowledgements

This work was supported by Australia Research Council under Discovery Project No. DP0558854.

References

1. G. Antoniou, D. Billington, G. Governatori, and M. Maher, Representation Results for Defeasible Logic, *ACM Transactions on Computational Logic*, 2(2), pp. 255-287, 2001.
2. D. Billington, Defeasible Logic is Stable, *Journal of Logic and Computation*, 3, pp. 370-400, 1993.
3. G. Governatori, Defeasible Description Logics, In G. Antoniou and H.Boley, editors, *Proceedings of Rules and Rule Markup Languages for the Semantic Web: Third International Workshop (RuleML 2004)*, pp. 98-112, 2004.
4. M. J. Maher, Propositional Defeasible Logic has Linear Complexity, *Theory and Practice of Logic Programming*, 1(6), pp. 691-711, 2001.
5. M. J. Maher, A. Rock, G. Antoniou, D. Billignton, and T. Miller, Efficient Defeasible Reasoning Systems, *International Journal of Artificial Intelligence Tools*, 10(4), pp. 483-501, 2001.
6. D. Nute, Defeasible Logic, In *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 3, pp. 353-395, Oxford University Press, 1987.
7. P. Pothipruk and G. Governatori, A Formal Ontology Reasoning with Individual Optimization: A Realization of the Semantic Web, In M. Kitsuregawa, E. Neuhold and A. Ngu, editors, *Web Information Systems Engineering (WISE 2005)*, pp. 119-132, 2005.