Conservative Retractions of Propositional Logic Theories by Means of Boolean Derivatives: Theoretical Foundations

Gonzalo A. Aranda-Corral[1], Joaquín Borrego-Díaz[1], and M. Magdalena Fernández-Lebrón[2]

[1] Departamento de Ciencias de la Computación e Inteligencia Artificial
[2] Departamento de Matemática Aplicada I
E.T.S. Ingeniería Informática, Universidad de Sevilla,
Avda. Reina Mercedes s.n. 41012-Sevilla, Spain
{garanda,jborrego,lebron}@us.es

Abstract. We present a specialised (polynomial-based) rule for the propositional logic called the *Independence Rule*, which is useful to compute the conservative retractions of propositional logic theories. In this paper we show the soundness and completeness of the logical calculus based on this rule, as well as other applications. The rule is defined by means of a new kind of operator on propositional formulae. It is based on the *boolean derivatives* on the polynomial ring $\mathbb{F}_2[\mathbf{x}]$.

Keywords: Conservative retraction, Independence Rule, boolean derivatives.

1 Introduction

A theory T is a conservative extension of a theory T' (or T' is a *conservative retraction*) if every consequence of T in the language of T' is a consequence of T' already. Conservative extensions have been deeply investigated in Mathematical Logic, and they allow to formalize several notions concerning refinements and modularity in Computer Science (for example, in formal verification [20,1,21]). In this paper we investigate how to compute a conservative retraction of a theory. In particular, we are interested in the following problem:

Conservative Retraction Problem (CRP):
- Input: A finite theory T in a language L, and $L' \subseteq L$.
- Output: A conservative retraction of T to the language L'.

Given a sublanguage L' of the language of T, a conservative retraction on L' has two basic properties:

⋆ Partially supported by Minerva -Services in Mobility Platform- Project *WeTeVe* (2C/040) and *Ayudas a grupos de investigación, Junta de Andalucía* (TIC 137).

– There always exists a conservative retraction of T. For example, such a theory is

$$\{F \in \text{Form}(L') \ : \ T \models F\} \qquad (\dagger)$$

– Any two conservative retractions of T in the same sublanguage are equivalent theories.

We will denote by $[T, L']$ a conservative retraction of T to the sublanguage L' throughout the paper.This paper is concerned with the problem of computing (finitely axiomatized) conservative retractions. The importance of the computing of conservative retractions, in any logic, is based on its potential applications. For example:

– *Location principle for Knowledge Based Systems (KBS) reasoning*: Suppose that KB is a knowledge base, and let F be a formula. Suppose also that the language of F is L'. The question

$$KB \overset{?}{\models} F$$

can be solved in two steps:

- A conservative retraction $[KB, L']$ has to be computed
- We have to decide whether $[KB, L'] \models F$

Note that the second question usually has lower complexity than the original one, due to relatively small size of L'. This observation is extremely interesting when KB is a huge ontology.

– It is usual to approach the retraction by means of syntactic analysis, in order to locate the reasoning on certain axioms ([23]). In these cases, the conservative retraction would be very useful.
For example, let us consider the following ontology, in Propositional Description Logic (see [3] and [16] for details):

$$\Sigma = \begin{cases} \mathsf{Virus} \sqsubseteq \mathsf{Animal} \sqcup \mathsf{MobileEntity} \\ \mathsf{Mammals} \sqsubseteq \mathsf{Animal} \sqcap \mathsf{MobileEntity} \\ \mathsf{Animal} \sqsubseteq \neg \mathsf{Plant} \end{cases}$$

Suppose that we want to specialize the reasoning in the concepts {Virus, Mammals, Plants, Animal} (because the concept MobileEntity is not contained in the superconcept LivingBeing), but we do not want to lose any knowledge about these concepts originally entailed by Σ. Note that it could be very hard, or not possible, to transform the ontology to obtain the conservative retraction by a syntactic analysis. Since this case is a propositional description logic ontology, it is possible to apply the method presented in this paper, obtaining thus the conservative retraction

$$\partial_{\mathsf{MobileEntity}}(\Sigma) = \begin{cases} \mathsf{Animal} \sqsubseteq \neg \mathsf{Plant} \\ \mathsf{Mammals} \sqsubseteq \mathsf{Animal} \end{cases}$$

At higher levels of expressivity, one can observe that existing tools provide syntactic modularity, but no semantic modularity.

– *Contextual reasoning.* In a similar way as in the above example, the conservative retraction $[KB, L']$ ensures the maximality of context knowledge with respect to the ontology source.
  A similar problem, in the complex case of ontological reasoning in OWL, is the use of partitioning methods by means $\mathcal{E}$-connections ([15]). Indeed the partitioning to an $\mathcal{E}$-connection provides modularity benefits; it typically contains several "free-standing" components, that is, sub-KBs which do not "use" information from any other components (observation also made in [15]).
– In SAT-based planning, the number of propositional variables is bigger than the size of any formula. Since the formulas without variables of L' are not used for the computing of a conservative retraction (as we will see in this paper), then computing conservative retractions may be a good strategy to synthetize partial plans. Similar ideas can be applied to obtain better partition-based reasoning algorithms for propositional logics ([2]).
– With regard to the specific case of the use of Computer Algebra Systems (CAS) for reasoning with knowledge-based systems in real problems (see e.g. [19]), the rule presented in this paper has interesting features. The use of CAS is based on a faithful translation of logical formulas into polynomials on finite fields. The algebraic counterpart of the Independence Rule, in algebraic geometry terms, is a tool for projecting varieties in positive characteristics. This interpretation is very useful to design new applications of Gröbner basis to Knowledge Based Systems.

On the one hand, to the best of our knowledge, there is no calculus specifically focused on the computing of conservative retractions. The main reason for this is that the notion of *conservative extension* is more interesting (for example in incremental specification/verification of systems). For instance, the Isabelle and ACL2 theorem provers adopt this methodology by providing a language for conservative extensions by definition (even for the specification and verification of the logic itself, see e.g. [1]). Another example are the formal approaches to Ontological reasoning and extending (see e.g. the conservative extensions generated by definitional methodologies [6]). And finally, weaker notions than conservative extensions are used in methods for ontological extensions assisted by automated reasoning systems (see [8,9]).

On the other hand, although the conservative retraction of theories can be interesting itself, in expressive logics (like first order logic) the retraction may not be finitely axiomatized (for example, in first order theories of arithmetic). It is even possible that it involves undecidable problems. In the concrete case of propositional logics, computing conservative retractions are feasible. One can, for example, translate the theory to clauses and then select a conservative retraction from the saturation by resolution of the clausal translation.

The main contribution of this paper is a new propositional rule, called *Independence Rule*, specifically designed to compute (and to deal with) conservative retractions. This is the first tool designed for effectively computing conservative retractions. The Independence Rule allows the systematic elimination of

propositional variables outside the sublanguage preserving, at the same time, the logical consequences in the sublanguage. Moreover, the rule is also useful to deal with other propositional logical problems, as it will be described.

Finally, it is necessary to note that the theoretical existence shown in (†) does not illustrate how to obtain a finite axiomatization of the conservative retraction. The method presented in this paper outputs a finite axiomatization of $[T, L']$.

The paper is organized as follows. The next section reviews the relationship between propositional logic and the ring $\mathbb{F}_2[\mathbf{x}]$. In the third section the *boolean derivatives* are introduced. Section 4 shows the soundness and completeness of a complete calculus based on them. Section 5 presents basic properties of the rule which are useful to simplify the computing. In section 6 we formalize the location principle as a basis for the computing of the conservative retraction. Section 7 is devoted to show other interesting applications of the Independence Rule, such as theory merging and conservative extensions built by hierarchical merging. We conclude with some remarks about future work.

2 Propositional Logic and the Ring $\mathbb{F}_2[\mathbf{x}]$

The algebraic translation of Propositional Logic into Polynomial Algebra is based on a well known translation of propositional logic in this kind of algebras (see [18], and also [12]). There exist several approaches and applications of this traslation, which allow the use of algebraic tools (as Gröbner Basis) for solving logical problems (see e.g. [5,13,10] and the application given in [19]). This section is devoted to review the main features.

We fix a propositional language $PV = \{p_1, \ldots, p_n\}$, $PForm$ denotes the set of propositional formulas in this language, and $var(F)$ denotes the set of variables of F.

The ring we work on is $\mathbb{F}_2[\mathbf{x}]$ (where $\mathbf{x} = x_1, \ldots, x_n$). A key ideal is $\mathbb{I}_2 := (x_1 + x_1^2, \ldots, x_n + x_n^2)$. To clarify the reasoning, we fix an identification $p_i \mapsto x_i$ (or $p \mapsto x_p$) between PV and the set of indeterminates.

Given $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, let us define $|\alpha| := \max\{\alpha_1, \ldots, \alpha_n\}$, and $sg(\alpha) := (\delta_1, \ldots, \delta_n)$, where δ_i is 0 if $\alpha_i = 0$ and 1 otherwise. If $a(\mathbf{x}) \in \mathbb{F}_2[\mathbf{x}]$,

$$\deg_\infty(a(\mathbf{x})) := \max\{|\alpha| \ : \ \mathbf{x}^\alpha \text{ is a monomial of } a\},$$

and $\deg_i(a(\mathbf{x}))$ is the degree w.r.t. x_i. If $\deg_\infty(a(\mathbf{x})) \leq 1$, $a(\mathbf{x})$ is called a *polynomial formula*.

Three maps represent the standard starting point for the translation from propositional logic into $\mathbb{F}_2[\mathbf{x}]$:

– The *flatening* map $\Phi : \mathbb{F}_2[\mathbf{x}] \to \mathbb{F}_2[\mathbf{x}]$ is defined by

$$\Phi(\sum_{\alpha \in I} \mathbf{x}^\alpha) := \sum_{\alpha \in I} \mathbf{x}^{sg(\alpha)}$$

Note that Φ satisfies

$$\Phi(\mathbb{I}_2) = (0) \text{ and } \Phi(a \cdot b) = \Phi(\Phi(a) \cdot \Phi(b))$$

– The *polynomial interpretation* $P : PForm \to \mathbb{F}_2[\mathbf{x}]$ assigns a polynomial to each logical formula. This is achieved by assigning to each propositional variable p_i a monomial x_i and defining, for each connective, the function as follows:
  - $P(\bot) = 0, P(p_i) = x_i, P(\neg F) = 1 + P(F)$
  - $P(F_1 \wedge F_2) = P(F_1) \cdot P(F_2)$
  - $P(F_1 \vee F_2) = P(F_1) + P(F_2) + P(F_1)P(F_2)$
  - $P(F_1 \to F_2) = 1 + P(F_1) + P(F_1)P(F_2)$
  - $P(F_1 \leftrightarrow F_2) = 1 + P(F_1) + P(F_2)$
– The *propositional interpretation* $\Theta : \mathbb{F}_2[\mathbf{x}] \to PForm$ is defined by:
  - $\Theta(0) = \bot,\ \Theta(1) = \top,\ \Theta(x_i) = p_i$,
  - $\Theta(a \cdot b) = \Theta(a) \wedge \Theta(b)$, and
  - $\Theta(a + b) = \neg(\Theta(a) \leftrightarrow \Theta(b))$.

We have that

$$\Theta(P(F)) \equiv F \text{ and } P(\Theta(a)) = a$$

Since we shall be frequently applying $\Phi \circ P$, we take $\pi := \Phi \circ P$, called the *polynomial projection*.

Next we list some basic results that we will use later on.

Lemma 1. *Let $v : PV \to \{0, 1\}$ be a valuation with $v(p_i) = \delta_i$. Then for every $F \in PForm$, $v(F) = P(F)(\delta_1, \ldots \delta_n)$.*

From any subset X of $\mathbb{F}^n$ we can cook up an ideal $I(X)$, the ideal of polynomials vanishing on X. From any subset I of $\mathbb{F}_2[\mathbf{x}]$ we can cook up an algebraic set $V(I)$, the "vanishing set" of the ideal. The behaviour of the ideals of $\mathbb{F}_2[\mathbf{x}]$ is well known:

– If $A \subseteq (\mathbb{F}_2)^n$, then $V(I(A)) = A$,
– For every $\mathfrak{I} \in Ideals(\mathbb{F}_2[\mathbf{x}])$, it holds that $I(V(\mathfrak{I})) = \mathfrak{I} + \mathbb{I}_2$.

Therefore $F \equiv F'$ if and only if $P(F) = P(F') \pmod{\mathbb{I}_2}$ which is also equivalent to $\Phi \circ P(F) = \Phi \circ P(F')$.

The following theorem states the main relationship between propositional logic and $\mathbb{F}_2[\mathbf{x}]$:

Theorem 1. *The following conditions are equivalent:*

1. $\{F_1, \ldots, F_m\} \models G$.
2. $1 + P(G) \in (1 + P(F_1), \ldots, 1 + P(F_m)) + \mathbb{I}_2$.
3. $\mathtt{NF}(1 + P(G), \mathtt{GB}\,[(1 + P(F_1), \ldots, 1 + P(F_m)) + \mathbb{I}_2]) = 0$. *(where* GB *denotes* Gröbner basis*) and* NF *denotes* normal form.

3 Boolean Derivatives and Non-clausal Theorem Proving

Boolean derivative is a well known tool in Boolean Function Calculus (cf. [22]). We introduce here the operator on propositional formulas as a translation of the usual derivation on $\mathbb{F}_2[\mathbf{x}]$. Recall that a derivation on a ring R is a map $d : R \to R$ verifying:

1. $d(a+b) = d(a) + d(b)$
2. $d(a \cdot b) = d(a) \cdot b + a \cdot d(b)$

Definition 1. *A map $\partial : PForm \to PForm$ is a* boolean derivation *if there exists a derivation d on $\mathbb{F}_2[\mathbf{x}]$ such that the following diagram commutes:*

$$\begin{array}{ccc} PForm & \stackrel{\partial}{\to} & PForm \\ \pi \downarrow & \# & \uparrow \Theta \\ \mathbb{F}_2[\mathbf{x}] & \stackrel{d}{\to} & \mathbb{F}_2[\mathbf{x}] \end{array}$$

That is,

$$\partial = \Theta \circ d \circ \pi$$

If the derivation on $\mathbb{F}_2[\mathbf{x}]$ is $d = \frac{\partial}{\partial x_p}$, we denote ∂ as $\frac{\partial}{\partial p}$. This derivation has an interesting property : The formula $\frac{\partial}{\partial p}(F)$ represents the change of truth value of F if the truth value of p is changed (recall that $F\{p/G\}$ denotes the formula obtained by substitution of p by the formula G in F).

Proposition 1. $\frac{\partial}{\partial p}F \equiv \neg(F\{p/\neg p\} \leftrightarrow F)$.

Proof. It is easy to see that

$$\pi(F\{p/\neg p\})(\mathbf{x}) = \pi(F)(x_1, \ldots, x_p + 1, \ldots, x_n).$$

Since $\frac{\partial}{\partial x}a(x) = a(x+1) + a(x)$ holds for polynomial formulas, one has

$$P(\frac{\partial}{\partial p}F) = \frac{\partial}{\partial x_p} \circ \pi(F)(\mathbf{x}) = \pi(F)(x_1, \ldots, x_p + 1, \ldots, x_n) + \pi(F)(\mathbf{x})$$

hence

$$\frac{\partial}{\partial x_p} \circ \pi(F)(\mathbf{x}) = \Phi(P(F\{p/\neg p\}) + P(F)) = \pi(\neg(F\{p/\neg p\} \leftrightarrow F)).$$

By application of Θ we have that $\frac{\partial}{\partial p}F \equiv \neg(F \leftrightarrow F\{p/\neg p\})$.

An important feature of the boolean derivative above defined is that the value of $\frac{\partial}{\partial p}F$ with respect to a valuation does not depend on p. Thus, we can apply any valuation on $PV \setminus \{p\}$ to this formula. That is, since for polynomial formulas

$$\Theta(\frac{\partial}{\partial x}a) \equiv \frac{\partial}{\partial p}\Theta(a)$$

we can assume that

$$\frac{\partial}{\partial p}F := \Theta(\frac{\partial}{\partial x_p}\pi(F))$$

so $p \notin var(\frac{\partial}{\partial p}F)$.

Definition 2. *The Independence Rule (or ∂-rule) on polynomial formulas $a_1, a_2 \in \mathbb{F}_2[\mathbf{x}]$ is defined as:*

$$\partial_x(a_1, a_2) := \quad \frac{a_1,\ a_2}{1 + \Phi\left[(1 + a_1 \cdot a_2)(1 + a_1 \cdot \frac{\partial}{\partial x}a_2 + a_2 \cdot \frac{\partial}{\partial x}a_1 + \frac{\partial}{\partial x}a_1 \cdot \frac{\partial}{\partial x}a_2)\right]}$$

In terms of polynomial coefficents, if we write $a_i = b_i + x_p \cdot c_i$, with $\deg_{x_p}(b_i) = \deg_{x_p}(c_i) = 0$ $(i = 1, 2)$, then

$$\partial_{x_p}(a_1, a_2) := \quad \frac{b_1 + x_p \cdot c_1,\ b_2 + x_p \cdot c_2}{\Phi\left[1 + (1 + b_1 \cdot b_2)[1 + (b_1 + c_1)(b_2 + c_2)]\right]}$$

Note that the rule is symmetric. The Independence Rule on formulas is defined by translating the above rule to formulas:

$$\partial_p(F_1, F_2) := \Theta(\partial_{x_p}(\pi(F_1), \pi(F_2))).$$

This is the propositional interpretation of the result of applying the (polynomial) independence rule to the polynomial projection of the formulas.

For example,

$$\begin{aligned}\partial_b(c \to a \vee b, d \to a \wedge b) &= \Theta[\partial_b(1 + c(1 + a)(1 + b),\ 1 + d(1 + ab))] \\ &= \neg c \vee a \vee \neg d\end{aligned}$$

Lemma 2. *Let $F \in PForm$ and p be a propositional variable. There exists $F_0 \in PForm$, such that $p \notin var(F_0)$ and*

$$F \equiv \neg(F_0 \leftrightarrow p \wedge \frac{\partial}{\partial p}F)$$

Proof. Consider the polynomial formula $a = \pi(F)$. Since $\deg_{x_p}(a) \leq 1$, there exists $b \in \mathbb{F}_2[\mathbf{x}]$ such that $\deg_{x_p}(b) = 0$ and $a = b + x_p \cdot \frac{\partial}{\partial x_p}a$.

By applying Θ, we conclude that

$$F \equiv \Theta(b + x_p \cdot \frac{\partial}{\partial x_p}a) \equiv \neg(\Theta(b) \leftrightarrow p \wedge \frac{\partial}{\partial p}F)$$

4 Soundness and Completeness of Independence Rule

The Independence Rule induces a concept of proof in the standard way, that we denote as $\vdash_\partial$.

Proposition 2. *The Independence Rule is sound.*

Proof. It is sufficent to see that $\{F_1, F_2\} \models \partial_p(F_1, F_2)$. If $\pi(F_i) = a_i$, with $a = b_i + x_p \cdot c_i$ and $\deg_{x_p}(b_i) = \deg_{x_p}(c_i) = 0$, then $F_i \equiv \neg[\Theta(b_i) \leftrightarrow p \wedge \Theta(c_i)]$ $(i = 1, 2)$.

Assume that v satisfies $v(F_1) = v(F_2) = 1$. If $v(p) = 1$ then

$$v(\neg[\Theta(b_i) \leftrightarrow \Theta(c_i)]) = 1;$$

if $v(p) = 0$ then $v(\Theta(b_i)) = 1$. Both cases imply that

$$\{F_1, F_2\} \models \bigwedge_{i=1,2} \neg[\Theta(b_i) \leftrightarrow \Theta(c_i)] \vee \bigwedge_{i=1,2} \Theta(b_i)$$

By application of π to the right-hand formula, and knowing that $c_i = \frac{\partial}{\partial x_p}\pi(F_i)$, one obtains the result by rewriting.

As seen in the above proof, $\deg_i(\partial_{x_i}(a_1, a_2)) = 0$, and the valuations are considered with respect to every possible value on p. Therefore, it is straightforward to prove the following property:

Corollary 1. *Let $v : PV\backslash\{p\} \to \{0, 1\}$. The following conditions are equivalent:*

1. *$v \models \partial_p(F_1, F_2)$.*
2. *Some extension of v to PV is a model of $\{F_1, F_2\}$.*

For example, consider the propositional formula $p_1 \wedge \neg p_2$. It has that

$$\pi(p_1 \wedge \neg p_2) = x_1(1 + x_2)$$

We have that

$$\partial_{x_1}(x_1(1 + x_2), x_1(1 + x_2)) = 1 + x_2,$$

so the valuation v such that $v(\neg p_2) = 1$ is the only one that we can extend to a model of $p_1 \wedge \neg p_2$. In the case of that $\partial_p(\pi(F_1), \pi(F_2)) = 1$, every partial valuation is extendable to a model of $\{F_1, F_2\}$. Analogously, if $\partial_p(\pi(F_1), \pi(F_2)) = 0$, then there is no valuation extendable to a model of both formulas.

The refutation procedure can be applied to formulas or their equivalent polynomials formulas. Let us see an example. An ∂-refutation for the set $\pi[\{p \to q, q \vee r \to s, \neg(p \to s)\}]$ is

1. $1 + x_1 + x_1x_2$	$[\![\pi(p \to q)]\!]$
2. $1 + (x_2 + x_3 + x_2x_3)(1 + x_4)$	$[\![\pi(q \vee r \to s)]\!]$
3. $x_1(1 + x_4)$	$[\![\pi(\neg(p \to s)]\!]$
4. $1 + x_1 + x_3 + x_1x_4 + x_3x_4 + x_1x_3 + x_1x_3x_4$	$[\![\partial_{x_2}$ *to* $(1), (2)]\!]$
5. 0	$[\![\partial_{x_1}$ *to* $(3), (4)]\!]$

The following theorem states the refutational completeness of ∂-rule:

Theorem 2. *If Γ is inconsistent then $\Gamma \vdash_\partial \bot$.*

Proof. Let $\partial_k[\Gamma]$ $(k \leq n)$ be the set of formulas defined by recursion as follows: $\partial_0[\Gamma] := \Gamma$ and, if $k \geq 1$,

$$\partial_k[\Gamma] := \{\partial_{p_k}(F_1, F_2) \ : \ F_1, F_2 \in \partial_{k-1}[\Gamma]\}$$

Note that if $F \in \partial_k[\Gamma]$, then $\mathrm{var}(F) \subseteq \{p_{k+1}, \ldots p_n\}$. Thus $\partial_n[\Gamma] \subseteq \{\top, \bot\}$. Therefore it is sufficent to prove that Γ is inconsistent if and only if $\bot \in \partial_n(\Gamma)$.

Since the rule is sound, if $\bot \in \partial_n[\Gamma]$, the set Γ has no models.

Assume now that $\partial_n[\Gamma] = \{\top\}$. Then the constant valuation 1 is a model of $\partial_n[\Gamma]$. By applying induction on k up to 0, it is sufficent to prove that one can extend a model of $\partial_k[\Gamma]$ to a model of $\partial_{k-1}[\Gamma]$.

Let $v : \{p_{k+1}, \ldots, p_n\} \to \{0, 1\}$ be a model of $\partial_k[\Gamma]$, and assume that v can not be extended to a model of $\partial_{k-1}[\Gamma]$. That is, if $v_i = v \cup \{(p_k, i)\}$, then there exists $F^i \in \partial_{k-1}[\Gamma]$ such that $v_i(F^i) = 0$ $(i = 0, 1)$. Note that $v_i(\partial_{p_k}(F^0, F^1)) = 1$ $(i = 1, 2)$.

By rewriting F^i as in lemma 2,

$$F^i \equiv \neg(F_0^i \leftrightarrow p_k \wedge \frac{\partial}{\partial p_k} F^i).$$

We conclude then that $v_0(F_0^0) = 0$, and hence $v(F_0^0) = 0$. Furthermore,

$$v(\neg(F_0^1 \leftrightarrow \frac{\partial}{\partial p_k} F^1)) = v_1(\neg(F_0^1 \leftrightarrow p \wedge \frac{\partial}{\partial p_k} F^1)) = 0.$$

Both facts imply that $v(\partial_{p_k}(F^0, F^1)) = 0$, leading to a contradiction, because $v \models \partial_k[\Gamma]$.

Applying induction, a model of $\partial_0[\Gamma] = \Gamma$ can be found.

The above proof suggests how to find models of Γ (when it is consistent). The decision procedure sketched in the proof is based on the partial saturation of Γ by the ∂-rule. Therefore the method can have a high cost, $O(|\Gamma|^{2^n})$.

5 Properties of the Independence Rule

The following result lists some basic properties that facilitate the computations:

Proposition 3. *Let F, G be propositional formulas*

1. $\partial_p(p, F) \equiv F\{p/\top\}$
2. *If* $p \notin var(F)$ *then* $\partial_p(F, G) \equiv F \wedge \partial_p(G, G)$
3. *If* $p \notin var(F) \cup var(G)$ *then* $\partial_p(F, G) \equiv F \wedge G$
4. $\partial_p(G, G) \equiv G\{p(\bot)\} \vee \neg(G\{p/\top\} \leftrightarrow G\{p/\bot\})$
5. $\partial(F_1 \wedge F_2, F_3) \equiv \partial_p(F_1, F_2 \wedge F_3)$
6. $\partial_p(F_1 \vee F_2, F_3) \equiv \partial_p(F_1, F_3) \vee \partial_p(F_2, F_3)$
7. $\partial_p(F_1, F_2) \equiv \partial_p(F_2, F_1)$

Proof. The proofs are based on algebraic manipulation of polynomial translation, except property (4), which follows from corollary 1.

Entailment can also be reduced by means of the Independence Rule:

Proposition 4.

$$\Gamma \models G \quad \Longrightarrow \quad \partial_p[\Gamma] \models \partial_p(G)$$

6 Location Principle as Conservative Retraction of Theories

Given $Q = \{q_1, \ldots, q_k\} \subseteq PV$ the operator $\partial_Q := \partial_{q_1} \circ \cdots \circ \partial_{q_k}$ is well defined modulo logical equivalence. This follows from corollary 1, because for every $p, q \in PV$,

$$\partial_p \circ \partial_q[\Gamma] \equiv \partial_q \circ \partial_p[\Gamma]$$

A consequence of corollary 1 and theorem 2 (its proof) is that entailment problem can be reduced to another one where only appears the variables of the goal:

Corollary 2. *(Location principle)* $\Gamma \models F \iff \partial_{PV \setminus var(F)}[\Gamma] \models F$

Proof. If $\Gamma \models F$ then $\partial_{PV \setminus var(F)}[\Gamma] \cup \{F\}$ is inconsistent (if not, a model of this set can be extended to a model of $\Gamma \cup \{F\}$ by corollary 1). The other implication is true because $\Gamma \models \partial_{PV \setminus var(F)}[\Gamma]$.

The corollary states that $\partial_{PV \setminus L'}[\Gamma]$ is an conservative retraction of Γ to L' (an instance of $[\Gamma, L']$). Thus, CRP problem is solved in this way for propositional logic.

From here, to simplify the notation, we identify $[\Gamma, L']$ with $\partial_{PV \setminus L'}[\Gamma]$.

7 Theory Merging and Hierarchical Theory Merging

In this section we describe how the Independence Rule can be used for theory merging. The following theorem can be considered a version of Craig's Interpolation Lemma for conservative retractions:

Theorem 3. *Let T_1 and T_2 be consistent theories with languages L_1 and L_2 respectively. The following conditions are equivalent:*

1. *$T_1 \cup T_2$ is consistent.*
2. *$[T_1, L_1 \cap L_2] \cup [T_2, L_1 \cap L_2]$ is consistent.*

Proof. (1) $\Longrightarrow$ (2) follows from the soundness of the Independence Rule, because a model of $T_1 \cup T_2$ is model of both retractions.

(2) $\Longrightarrow$ (1) follows from the completeness of the Independence Rule: if

$$v \models [T_1, L_1 \cap L_2] \cup [T_2, L_1 \cap L_2]$$

then there exists two extensions of v, v_1 and v_2, such that $v_1 \models T_1$ and $v_2 \models T_2$. Since the common variables to L_1 and L_2 are in the domain of v, we have that $v_1 \cup v_2$ is a well defined valuation which models $T_1 \cup T_2$.

The above theorem establishes a necessary and sufficent condition for theory merging. However, there are some situations where the merging is inconsistent but it would be interesting to extend one of the theories with consistent knowledge entailed by the other one. For example, when we aim to merge ontologies which have uncertain concepts.

Consider the ontology

$$\Sigma' = \begin{cases} \mathsf{Bacteria} \sqsubseteq \mathsf{Animal} \sqcup \mathsf{MobileEntity} \\ \mathsf{Fish} \sqsubseteq \mathsf{Animal} \sqcap \mathsf{MobileEntity} \\ \mathsf{MobileEntity} \sqsubseteq \neg \mathsf{Mammals} \end{cases}$$

It has that $\Sigma \cup \Sigma'$ entails $\mathsf{Mammals} \equiv \bot$, thus the union is inconsistent. However, it is feasible to extend Σ with knowledge from Σ'. The idea is to retract the second theory to interesting concept symbols, for example to the set

$$\{\mathsf{Bacteria}, \mathsf{Fish}, \mathsf{Animal}, \mathsf{Mammals}\}$$

In this case, the resultant ontology is consistent:

$$\Sigma \cup \partial_{\mathsf{MobileEntity}}(\Sigma') = \begin{cases} \mathsf{Virus} \sqsubseteq \mathsf{Animal} \sqcup \mathsf{MobileEntity} \\ \mathsf{Mammals} \sqsubseteq \mathsf{Animal} \sqcap \mathsf{MobileEntity} \\ \mathsf{Animal} \sqsubseteq \neg \mathsf{Plant} \\ \mathsf{Bacteria} \sqsubseteq \mathsf{Animal} \\ \mathsf{Fish} \sqsubseteq \mathsf{Animal} \end{cases}$$

It is also possible for the ontology obtained in this way to be inconsistent. The following result shows a case in which the extension of the ontology source is consistent:

Lemma 3. *Let T_1 and T_2 be consistent theories in the languages L_1 and L_2, respectively. The theory $T_1 \cup \partial_{L_1 \cap L_2}(T_2)$ is consistent*

In order to formalize the above ideas, we introduce the notion of *hierarchical merging*.

Definition 3. *Let T_1 and T_2 be consistent theories in the languages L_1 and L_2, respectively. A* hierarchical merging *of T_1 and T_2, is a theory T such that:*

1. *T is a conservative extension of T_1.*
2. *For any formula F in the language of $L_2 \setminus L_1$,*

$$T \models F \iff T_2 \models F$$

3. *Whenever theory T' satisfies (1) and (2), $T' \models T$ is verified.*

Thus, the Independence Rule is useful to show that the hierarchical merging of two theories is unique modulo equivalence (when it exists). The result is straightforward from the properties of ∂:

Theorem 4. *Under the conditions of the above definition, $T_1 \cup \partial_{L_1 \cap L_2}(T_2)$ is a hierarchical merging of T_1 and T_2.*

8 Related Work, Conclusions and Future Work

A related rule is the *general resolution* (cf. [4]):

$$Res_p(F,G): \qquad \frac{F,\ G}{F\{p/\top\} \vee G\{p/\bot\}}$$

(although it is expressed with respect to propositional variables, the original rule allows substitution of any subformula). For polynomial formulas $a_1, a_2 \in \mathbb{F}_2[\mathbf{x}]$ the rule is translated as follows:

$$Res_x(a_1,a_2): \qquad \frac{a_1,\ a_2}{\Phi(1+(1+a_1+(x+1)\frac{\partial}{\partial x}a_1)(1+a_2+x\frac{\partial}{\partial x}a_2))}$$

The general resolution is sound and refutationally complete. It is easy to see that

$$\models \partial_x(F,G) \rightarrow Res_x(F,G)$$

but in general it is not an equivalence[1].

Throughout the paper we pointed out related work using similar tools to the used here. To the best of our knowledge, no work on algebraic methods applied to conservative retraction was ever done. However, it is possible to use the elimination theorem on Gröbner basis in order to obtain a conservative retraction (see [14]). However, the elimination of polynomial variables depends on the selected lex ordering on variables for computing the Gröbner basis.

The future work may follow two lines. The first one is the extension to many-valued logics and their applications (see e.g. [19]). For this, a careful generalization of boolean derivatives, with nice logical meaning, seems necessary (in that case, it seems interesting to use another kind of derivations on polynomials on finite fields, as for example the Hasse-Schmidt derivations, see [17]). In the short term we are working on the extensions of ∂_p-rule to certain Description Logics with limited expressivity (as EL logic, [20], and some members of the $DL-lite$ family of Description Logics, see [11]), as well as the use of this rule for solving problems about definability in these logics.

References

1. Alonso, J.-A., Borrego-Díaz, J., Hidalgo, M.-J., Martín-Mateos, F.-J., Ruiz-Reina, J.-L.: A Formally Verified Prover for the ALC Description Logic. In: Schneider, K., Brandt, J. (eds.) TPHOLs 2007. LNCS, vol. 4732, pp. 135–150. Springer, Heidelberg (2007)
2. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artificial Intelligence 162(1-2), 49–88 (2005)
3. Baader, F., Calvanese, D., McGuinnes, D.L., Nardi, P., Patel-Schneider, P.F.: The Description Logics Handbook. Theory, Implementations and Applications. Cambridge University Press, Cambridge (2003)

[1] This fact also implies refutational completeness for ∂-rule. We showed the proof of th. 2 to remark how to find models and to explain the role of the operator $\partial_p[.]$.

4. Bachmair, L., Ganzinger, H.: A theory of resolution. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. I, pp. 19–99. Elsevier Science Pub., Amsterdam (1998)
5. Beame, P., Impagliazzo, R., Krajícek, J., Pitassi, T., Pudlák, P.: Lower Bounds on Hilvert's Nullstellensatz and propositional proofs. Proc. of London Mathematical Society 73, 1–26 (1996)
6. Bennett, B.: Relative Definability in Formal Ontologies. In: Proc 3rd Int. Conf. Formal Ontology in Information Systems (FOIS 2004), pp. 107–118. IOS Press, Amsterdam (2004)
7. Bochmann, D., Posthoff, C.: Binäre dynamishe systeme. Akademieverlag, Berlin (1981)
8. Borrego-Díaz, J., Chávez-González, A.M.: Extension of ontologies assisted by automated reasoning systems. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 247–253. Springer, Heidelberg (2005)
9. Borrego-Díaz, J., Chávez-González, A.M.: Controlling ontology extension by uncertain concepts through cognitive entropy. In: Proc. Workshop ISWC05 Uncertainty Reasoning on the Semantic Web URSW 2005, pp. 56–66 (2005), `http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/`
10. Buresh-Oppenheim, J., Clegg, M., Imppagliazzo, R., Pitassi, T.: Homogeneization and the Polynomial Calculus. Computational Complexity 11, 91–108 (2003)
11. Calvanese, D., De Giacomo, G., Lombo, D., Lenserini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The $DL-Lite$ Family. J. Automated Reasoning 39, 385–429 (2007)
12. Chazarain, J., Alonso-Jiménez, J.A., Briales-Morales, E., Riscos-Fernández, A.: Multi-valued logic and Gröbner bases with applications to modal logic. Journal Symbolic Computation 11, 181–194 (1991)
13. Clegg, M., Edmonds, J., Impagliazzo, R.: Using Gröbner Basis algorithm to find proofs of unsatisfiability. In: Proc. ACM Symposium of Computing, pp. 174–183 (1996)
14. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, Heidelberg (2005)
15. Cuenca-Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Automatic Partitioning of OWL Ontologies Using E-Connections. In: Proc. 2005 Int. Workshop on Description Logics (DL2005) (2005), `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-147/21-Grau.pdf`
16. Giunchiglia, F., Yatskevitch, M., Shvaiko, P.: Semantic Matching: Algorithms and Implementations. J. Data Semantics 9, 1–38 (2007)
17. Fernández-Lebrón, M., Narváez-Macarro, L.: Hasse-Schmidt Derivations and Coefficient Fields in Positive Characteristics. J. of Algebra 265(1), 200–210 (2003)
18. Kapur, D., Narendran, P.: An equational approach to theorem proving in first-order predicate calculus. In: Proc. 9 Int. Joint Conf. on Artificial Intelligence (IJCAI 1985), pp. 1146–1153 (1985)
19. Laita, L.M., Roanes-Lozano, E., de Ledesma, L., Alonso-Jiménez, J.A.: A computer algebra approach to verification and deduction in many-valued knowledge systems. Soft Computing 3, 7–19 (1999)

20. Lutz, C., Wolter, F.: Conservative extensions in the lightweight description logic EL. In: Pfenning, F. (ed.) CADE 2007. LNCS, vol. 4603, pp. 84–99. Springer, Heidelberg (2007)
21. Martín-Mateos, F.J., Alonso, J.A., Hidalgo, M.J., Ruiz-Reina, J.L.: Formal Verification of a Generic Framework to Synthetize SAT-Provers. J Aut. Reasoning 32(4), 287–313 (2004)
22. Thayse, A.: Boolean Calculus of Differences. Springer, Berlin (1981)
23. Tsarkov, D., Horrocks, I.: Optimised Classification for Taxonomic Knowledge Bases. In: Proc. 2005 Int. Workshop on Description Logics (DL 2005) (2005), http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-147/39-TsarHorr.pdf