Pietro Torasso (Ed.)

Advances in Artificial Intelligence

Third Congress of the Italian Association for Artificial Intelligence, AI*IA '93
Torino, Italy, October 26-28, 1993
Proceedings

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editor

Jörg Siekmann
University of Saarland
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3,
D-66123 Saarbrücken, Germany

Volume Editor

Pietro Torasso
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, I-10149 Torino, Italia

CR Subject Classification (1991): I.2

ISBN 3-540-57292-9 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57292-9 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Typesetting: Camera ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Preface

This book contains 22 long papers and 13 short ones which have been selected for the Scientific Track of the Third Congress of the Italian Association for Artificial Intelligence. Long papers are intended to report completed work, whereas short papers are mainly devoted to ongoing research. The Program Committee has strictly enforced the rule that only original and unpublished work can be considered for inclusion in the Scientific Track.

The papers report on significant work carried out in the different subfields of Artificial Intelligence, not only in Italy, but also in other European countries as well as outside Europe. Although the congress is organized by the Italian Association for Artificial Intelligence, it has a truly international character because of the invited speakers (Prof. Tom Mitchell, CMU, USA, Prof. Jean-Paul Barthes, Université de Technologie de Compiegne, France, Dr. Bernhard Nebel, DFKI, Germany), the number of papers presented by foreign authors, and the large number of submissions (roughly 40% of the total) coming from abroad.

The Program Committee had a hard job in evaluating the manuscripts submitted for publications since for most papers three independent reviews have been obtained (in some cases four).

Therefore, we believe that the book is a relevant source of information for understanding which are the currently active areas of research and the new promising directions in the AI field. Even if a single book cannot provide a complete picture of what is going on in AI (for example the areas of Perception and Vision, Qualitative Reasoning and Distributed Artificial Intelligence are somewhat underrepresented with respect the amount of activity carried on in Italy), some directions can be singled out.

Areas such as Automated Reasoning, Knowledge Representation and Natural Language (which have a well-established tradition in Italy) continue to attract significant amount of interest.

Machine Learning has recently attracted a lot of attention (not only among Italian scientists): the area has matured rapidly and a variety of approaches are currently being investigated, ranging from logical approaches (such as in Inductive Logic Programming) to numeric ones (as in genetic algorithms). This variety of approaches is well documented in the papers collected in the book.

Connectionism (or, more generally, subsymbolic approaches) has recently attracted significant interest within the AI community. In the book the application of subsymbolic approaches to perception and vision as well as to quite different problems is documented. Moreover, a increasing attention is being paid to the mechanisms for integrating symbolic and subsymbolic methods.

Inspecting the contents of the book, a growing interest for an explicit representation of time is apparent. The capability of developing an explicit representation of time and the need of performing temporal reasoning in an efficient way is relevant not only in the area of knowledge representation, but also in planning, robotics and reasoning about physical systems.

In achieving the goal of organizing a congress of high scientific level, the contribution and the efforts of many persons have to be acknowledged: beside authors, the Program Committee members and the referees (whose names are listed in the following pages) deserve my gratitude.

The financial support by Consiglio Nazionale delle Ricerche (Comitato Scienze d'Ingegneria e Architettura e Comitato Scienze e Tecnologia dell'Informazione) for partially covering the publication cost of the book is acknowledged.

Torino, July 1993

Piero Torasso
AI*IA '93 Program Chairman

Program Chairman

Pietro TORASSO (Università di Torino)

Program Committee - Scientific Track

Giovanni ADORNI (Università di Parma)
Luigia AIELLO (Università di Roma)
Amedeo CAPPELLI (CNR - Pisa)
Cristiano CASTELFRANCHI (CNR - Roma)
Mauro DI MANZO (Università di Genova)
Floriana ESPOSITO (Università di Bari)
Salvatore GAGLIO (Università di Palermo)
Attilio GIORDANA (Università di Torino)
Nicola GUARINO (CNR - Padova)
Francesco LAURIA (Università di Napoli)
Alberto MARTELLI (Università di Torino)
Maria SIMI (Università di Pisa)
Marco SOMALVICO (Politecnico di Milano)
Oliviero STOCK (IRST - Trento)
Luigi STRINGA (IRST - Trento)
Carlo TASSO (Università di Udine)

Program Committee - Application Track

Franco CANEPA (Alenia - Torino)
Maurizio DEL CANTO (Elsag Bailey - Genova)
Giannetto LEVIZZARI (Centro Ricerche FIAT - Orbassano)
Paolo MISSIO (Assicurazioni Generali - Mogliano Veneto)
Renato PETRIOLI (Fondazione Bordoni - Roma)
Roberto SERRA (Ferruzzi Finanziaria - Ravenna)
Luca SPAMPINATO (Quinary - Milano)

Organizing Committee

Cristina BENA (CSI-Piemonte - Torino)
Danilo DABBENE (Alenia - Torino)
Lorenza SAITTA (Università di Torino)
Nunzio SORRENTINO (Centro Ricerche FIAT - Orbassano)

Referees

Albesano
Amati G.
Ardizzone E.
Attardi B.
Atzeni P.
Avesani P.
Badaloni S.
Bagnara R.
Bergadano F.
Biagioli C.
Bisiani R.
Boldrin L.
Bonatti P.
Botta M.
Braggiotti A.
Brajnik G.
Broggi A.
Cadoli M.
Callari F.
Caprile B.
Carpineto C.
Caselli S.
Cesta A.
Chella A.
Chittaro L.
Cialdea Mayer M.
Cocco N.
Console L.
Conte R.
Crespi B.
Cristiani M.
D'Aloisi D.
D'Andrea V.
D'Angelo A.
De Giacomo G.
Delmonte R.
Donini F.
Fanelli A.M.
Federico M.
Ferrari G.
Fum D.
Furlanello C.
Gaspari M.
Gemello R.
Gerevini A.
Giaretta P.
Giordano L.
Giunchiglia E.
Lavelli A.
Lesmo L.
Lombardo V.
Magnini B.
Maier E.
Malerba D.
Mana F.
Martini S.
Miceli M.
Miola A.
Moretti L.
Nardi D.
Omodeo E.
Pedreschi D.
Pernici B.
Pirri F.
Pirrone R.
Poggi A.
Pogliano P.
Prodanof I.
Ricci F.
Roberto V.
Romano G.
Saitta L.
Sapino M.L.
Schaerf M.
Sebastiani R.
Semeraro G.
Serafini L.
Starita A.
Straccia U.
Strapparava C.
Tecchioli G.
Temperini M.
Terenziani P.
Toppano E.
Tornielli G.
Trautteur G.
Traverso P.
Vaggi A.
Varzi A.
Zaccaria R.
Zanichelli F.

Contents

Automated Reasoning

Cognitive Models

Connectionist Models and Subsymbolic Approaches

Knowledge Representation

Languages, Architectures and Tools for AI

Machine Learning

Proving Formulas through Reduction to Decidable Classes

Mauro Di Manzo
Enrico Giunchiglia
Alessandro Armando
Paolo Pecchiari

Mechanized Reasoning Group
DIST - University of Genoa
Via Opera Pia 11A, 16145 Genoa, Italy
{mauro,enrico,armando,peck}@dist.unige.it

Keywords: interactive theorem proving, decision procedures

Abstract. As it is well known, it is important to enrich the basic deductive machinery of an interactive theorem prover with complex decision procedures. In the GETFOL system we have implemented a hierarchical and modular structure of procedures which can be either invoked individually or jointly with the others. At the top of the hierarchy there is a decision procedure for a set of formulas which can be *reduced* to the class of prenex universal-existential formulas via finitely many application of rewriting rules. In this paper we give a formal account of such a reduction process, arguing that (*i*) it greatly enlarges the set of formulas which can proven through a decision process and (*ii*) in some cases makes the resulting formula easier to prove. We also provide an extensional characterization of a class of formulas which can be reduced and thus decided. The implementation of such reducing procedure in GETFOL is also sketched.

1 Introduction

Much of the work in interactive theorem proving deals with the definition of powerful and effective proof strategies. However, due to the simplicity of the basic inference steps, the design and synthesis of complex proof strategies may turn out to be a boring, hard and even unnatural activity. For example, in GETFOL [1], FOL [2] and LCF [3] it is neither easy nor natural to write a proof strategy for quantifier-free formulas basing on the rules provided by such systems (analogous to the rules for system of classical Natural Deduction as described in [4]).

A way to tackle this problem is to enrich the basic deductive machinery of the interactive theorem prover with complex decision procedures [5, 6, 7]. For example, in the GETFOL system we have implemented a hierarchical and modular structure of procedures which can be either invoked individually or jointly with the others [8, 9, 10]. At the top of the hierarchy there is a decision procedure for a set of formulas which can be *reduced* to the class of prenex universal-existential formulas (UE-formulas)[1] via finitely many application of truthful preserving rewriting rules.

Two are the main advantages of incorporating such a reduction process in an interactive theorem prover. First, supposing a decider for UE-class is already available (as in the GETFOL system), it significantly enlarges the class of formulas which can be proven through a decision process (see theorem 9). Second, in some cases the reduction process makes the resulting formula easier to prove (see example 1). This paper provides both a presentation of the theoretical properties of such a reduction process and a brief discussion of its implementation. While the former should give evidence of effectiveness of the procedures the latter should act as a more precise guideline for the understanding of how it is mechanized.

The procedures described in this paper have been implemented and are currently available inside the GETFOL system [1]. We want to recall that GETFOL provides the user with a set of inference rules which are very close to those of Natural Deduction [4]. In proving a theorem, it is possible to use only decision procedures (*e.g.* if the goal exactly matches the applicability conditions of the decision procedure), or to mix the application of inferences rules and decision procedures (*e.g.* for proving some sub-goals), or to use only inference rules (*e.g.* if decision procedures are not applicable or effective enough).

2 Enlarging the class of solved formulae

Consider the set $\mathcal{S}$ of rewriting rules (from now on $\mathcal{S}$-rules) expressing the well-known properties of associativity, commutativity and distributivity of the propositional connectives ($\mathcal{S}_2$-rules) and the distributivity of quantifiers through propositional connectives ($\mathcal{S}_1$-rules). Many formulae not in UE-form can be *reduced* to UE-formulae by finitely many applications of $\mathcal{S}$-rules. The reduce procedure implements the notion of reducibility w.r.t. $\mathcal{S}$. However, we want to point out that the notion of reducibility, upon which reduce has been built, is not bound to any particular set of rewriting rules. Hence the same methodology we used for building reduce can be used to build new procedures relying on other sets of rewriting rules.

The use of reduce greatly enlarges the class of formulae solved by the overall system. Here some examples of $\mathcal{S}$-reducible (to UE-form) formulae follow (If $\mathcal{C}$ is a set of rewriting rules then $\mapsto_{\mathcal{C}}$ is the reducibility relation induced by $\mathcal{C}$ and $\overset{*}{\mapsto}_{\mathcal{C}}$ is the reflexive and transitive closure of $\mapsto_{\mathcal{C}}$.)

[1] UE-formulae are formulae not containing function symbols and such that any universal quantifier does not contain free occurrences of existentially bounded variables in its scope. The UE-class is the set of UE-formulae. Obviously, the class of prenex Universal-Existential formulae not containing function symbol is contained in the UE-class.

Example 1.

(1) $\exists x.\forall y.(P(x,a) \vee R(y)) \quad \mapsto_{S_1} \exists x.(P(x,a) \vee \forall y.R(y))$

(2) $\exists x.\forall y.\exists z.(P(x,z) \vee P(y,z)) \quad \mapsto_{S_1} \exists x.\forall y.(\exists z_1.P(x,z_1) \vee \exists z_2.P(y,z_2))$
$\mapsto_{S_1} \exists x.(\exists z_1.P(x,z_1) \vee \forall y.\exists z_2.P(y,z_2))$

(3) $\exists x.\forall y.((P(y,a) \vee Q(x)) \vee Q(y)) \quad \mapsto_{S_2} \exists x.\forall y.((P(y,a) \vee Q(y)) \vee Q(x))$
$\mapsto_{S_1} \exists x.(\forall y.(P(y,a) \vee Q(y)) \vee Q(x))$

(4) $\exists x.\forall y.\exists z.((P(y,z) \wedge Q(x)) \wedge Q(z)) \mapsto_{S_2} \exists x.\forall y.\exists z.((P(y,z) \wedge Q(z)) \wedge Q(x))$
$\mapsto_{S_1} \exists x.\forall y.(\exists z.(P(y,z) \wedge Q(z)) \wedge Q(x))$
$\mapsto_{S_1} \exists x.(\forall y.\exists z.(P(y,z) \wedge Q(z)) \wedge Q(x))$

As the formula (1) in example 1 shows, the idea is to try to reduce the scope of universal quantifiers till they no longer contain free occurrences of existentially bounded variables. Formula (2) shows that in some case it is necessary also to consider the rules for pushing existential quantifiers. Formulas (3) and (4) evidence that in some cases to reduce the scope of an universal quantifier we have (first) to apply rules for a propositional manipulation of the matrix of a quantifier.

2.1 Basic definitions and theorems

This section is devoted to formally state and prove standard properties (*i.e.* noetherianity and confluence [11]) of the rewriting rules which we have informally spoken about. In order to provide a precise account of the rewriting rules used by **reduce** and to discuss their formal properties we introduce some notational conventions and definitions. $\alpha(x)$ denotes a formula in which there is at least one free occurrence of the variable x. $\alpha[x]$ denotes a formula in which there is no free occurrences of x. Q and Q' stand either for $\forall$ or for $\exists$. If $Q = \forall$, then $\circ = \wedge$ and $+ = \vee$. If $Q = \exists$, then $\circ = \vee$ and $+ = \wedge$.

Definition 1 minimality. A formula β is *minimal w.r.t.* $\langle Q,\ x\rangle$ if and only if satisfies one of the following clauses:

- (i) is a literal in which x occurs free,
- (ii) $\beta = Q'y.\gamma(x)$, with γ minimal w.r.t. $\langle Q',\ y\rangle$,
- (iii) $\beta = (\gamma(x) + \delta(x))$ with γ, δ minimal w.r.t. $\langle Q,\ x\rangle$.

A formula α is *minimal* if and only if each subformula $Qx.\beta$ of α is such that β is minimal w.r.t. $\langle Q,\ x\rangle$.

Definition 2 normalization. A formula β is *normalized w.r.t.* $\langle Q,\ x\rangle$ if and only if satisfies one of the following clauses:

- (i) is minimal w.r.t. $\langle Q,\ x\rangle$,

(*ii*) $\beta = \beta[x]$ (β does not contain free occurrences of x),
(*iii*) $\beta = (\gamma \circ \delta)$ with γ, δ normalized w.r.t. $\langle Q, x\rangle$,
(*iv*) $\beta = (\gamma[x] + \delta)$ with δ normalized w.r.t. $\langle Q, x\rangle$ (note that by clause (*i*) γ is normalized w.r.t. $\langle Q, x\rangle$).

A formula α is *normalized* if and only if each subformula $Qx.\beta$ of α is such that β is normalized w.r.t. $\langle Q, x\rangle$.

We also say that $\beta = (\gamma(x) + \delta)$ is a *top normalizable formula w.r.t.* $\langle Q, x\rangle$ if and only if γ and δ are normalized w.r.t. $\langle Q, x\rangle$ and β is not normalized w.r.t. $\langle Q, x\rangle$.

Given a formula α, β is a *top normalizable formula*[2] if and only if there exists $\langle Q, x\rangle$ such that:

(*i*) β is a top normalizable formula w.r.t. $\langle Q, x\rangle$,
(*ii*) descending α construction tree, $\langle Q, x\rangle$ are the last quantifier symbol and bound variable that we meet before β.

The rewriting rules used by **reduce** are listed in table 1. In the following we will refer to the sets of rules $\{(1)-(8)\}$, $\{(1)-(3)\}$ and $\{(4)-(8)\}$ by $\mathcal{S}$, $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. Notice that no rule in $\mathcal{S}$ is applicable to a minimal formula and that no rule in $\mathcal{S}_2$ is applicable to a normalized formula. The following two theorems establish the *noetherianity* and *confluence* of $\mathcal{S}$, $\mathcal{S}_1$ and $\mathcal{S}_2$.

(1)	$Qx.\alpha[x] \mapsto \alpha$
(2)	$Qx.(\alpha \circ \beta)(x) \mapsto (Qx.\alpha \circ Qx.\beta)$
(3)	$Qx.(\alpha[x] + \beta(x)) \mapsto (\alpha[x] + Qx.\beta(x))$
(4)	$(\alpha(x) + \beta[x]) \mapsto (\beta[x] + \alpha(x))$
(5)	$((\alpha[x] + \beta(x)) + \gamma(x)) \mapsto (\alpha[x] + (\beta(x) + \gamma(x)))$
(6)	$((\alpha \circ \beta)(x) + \gamma(x)) \mapsto ((\alpha + \gamma(x)) \circ (\beta + \gamma(x)))$
(7)	$(\alpha(x) + (\beta[x] + \gamma(x))) \mapsto (\beta[x] + (\alpha(x) + \gamma(x)))$
(8)	$(\alpha(x) + (\beta \circ \gamma)(x)) \mapsto ((\alpha(x) + \beta) \circ (\alpha(x) + \gamma))$

Restrictions:

- In rules $\{(4)-(8)\}$ the left hand side must be a top normalizable formula.
- In rules $\{(7), (8)\}$ α must be minimal w.r.t. $\langle Q, x\rangle$.

Table 1. The rewriting rules

[2] More precisely, we should say *"top normalizable formula occurrence in α"*.

Theorem 3. *$\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2$ are noetherian.*

Proof. For any formula α, define $\mathcal{F}(\alpha)$ to be the cardinality of the set of proper subformulae in the scope of a quantifier in α. As result of an application of one of the rules in $\{(1)-(3)\}$, $\alpha \mapsto \alpha'$ and $\mathcal{F}(\alpha') < \mathcal{F}(\alpha)$. Hence $\mathcal{S}_1$ is noetherian since $\{(1)-(3)\}$ can be applied only finitely many times to a formula. For any formula α, define $\mathcal{C}(\alpha)$ to be the cardinality of the set of proper subformulae of a top normalizable formula in α. Also note that to any top normalizable formula, one of rule in $\{(4)-(8)\}$ must be applicable. If α' is the formula obtained by applying such a rule then $\mathcal{C}(\alpha') < \mathcal{C}(\alpha)$. Hence also $\mathcal{S}_2$ is noetherian since $\{(4)-(8)\}$ can be applied only finitely many times. $\mathcal{S}$ is noetherian since, if we define $\mathcal{G}(\alpha) = (\mathcal{F}(\alpha) + 2 \times \mathcal{C}(\alpha))$ then the application of the rules in $\mathcal{S}$ makes $\mathcal{G}$ to decrease.

In defining $\mathcal{G}(\alpha)$, we give a different weight to $\mathcal{C}(\alpha)$ because if we apply rules (6) or (8), $\mathcal{F}(\alpha)$ can increase. With such weights, even for such rules, $\mathcal{G}(\alpha)$ is strictly decreasing.

Theorem 4. *$\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2$ are confluent.*

Proof. It is sufficient to notice that there are no critical pairs in $\mathcal{S}$, *i.e.* no two (variables disjoint) rewriting rules $(l_1 \mapsto r_1)$, $(l_2 \mapsto r_2)$ such that any (non variable) sub-term of l_1 is unifiable with l_2. Hence, also $\mathcal{S}_1, \mathcal{S}_2$ are trivially confluent.

2.2 Formal results about reducibility

We are now ready to discuss the properties of $\mathcal{S}$ with respect to the problem of reducing formulae in UE-form. As already said, the UE-reducibility of fairly wide classes of formulae is proved (theorem 9 and corollary 10). Such a result is a consequence of theorem 8 which states that indefinite applications of the rules in $\mathcal{S}$ have the effect to reduce the input formula to an equivalent minimal one. An effective way to accomplish such a reduction process is to recursively descend the formula tree and then apply the rewriting rules in a bottom up fashion (see procedure `S-normalize` in figure 1) exploiting the following facts:

- literals, conjunctions and disjunctions of minimal formulae are minimal,
- a minimal formula occurring in the scope of "Qx" can be rewritten into a normalized formula w.r.t. $\langle Q, x\rangle$ by applying the rules in $\mathcal{S}_2$ (procedure `S2-normalize` - lemma 7),
- A normalized formula (wrt $\langle Q, x\rangle$) can be turned into minimal form by application of the rules in $\mathcal{S}_1$ (procedure `S1-normalize` - lemma 5).

The `S-normalize` procedure, given the `S1-normalize` and `S2-normalize` procedures, (whose implementation is omitted for lack of space) has the effect to reduce the input formula `w` in minimal form. Before formally enunciating and proving theorems, the following example justifies the initial claim that in some cases the reduction process makes the formula easier to prove.

```
(DEFLAM S-normalize (w)
 (IF (LITERAL w) w
  (IF (CONJ w)
   (mkand (S-normalize (wff-get-lf w)) (S-normalize (wff-get-rt w)))
   (IF (DISJ w)
    (mkor (S-normalize (wff-get-lf w)) (S-normalize (wff-get-rt w)))
    (IF (QUANTWFF w)
     (S1-normalize
      (S2-normalize
       (mkquantwff (quantof w) (bvarof w) (S-normalize (matrix w)))))
     (ERRMESS "wff not in negative normal form"))))))
```

Fig. 1. The `S-normalize` routine.

Example 2. Problem 29 from [12].[3] The formula to be proven is:

$$((\exists x.F(x)\wedge\exists x.G(x))\rightarrow((\forall x.(F(x)\rightarrow H(x))\wedge\forall x.(G(x)\rightarrow J(x)))\leftrightarrow (\forall x.\forall y.((F(x)\wedge G(y))\rightarrow(H(x)\wedge J(y))))))$$

Applying **reduce** we get:

$$((\exists x.F(x)\wedge\exists x.G(x))\rightarrow((\forall x.(F(x)\rightarrow H(x))\wedge\forall x.(G(x)\rightarrow J(x)))\leftrightarrow ((\exists x.G(x)\rightarrow\forall x.(F(x)\rightarrow H(x)))\wedge (\exists x.F(x)\rightarrow\forall x.(G(x)\rightarrow J(x))))))$$

which can be proven using only propositional argumentations. For example, mapping each quantified formula into a distinct propositional letter, we obtain

$$((A\wedge B)\rightarrow((C\wedge D))\leftrightarrow((B\rightarrow C))\wedge(A\rightarrow D))$$

which is a tautology.

The following lemmas are needed to make the proof of theorem 8 easier.

Lemma 5. *If α is normalized w.r.t. $\langle Q,\, x\rangle$ then $Qx.\alpha \overset{*}{\mapsto}_{S_1} \beta$ with β minimal.*

Such a lemma easily follows from the definitions of formula normalized wrt $\langle Q,\, x\rangle$ and of minimality.

Lemma 6. *If α is a top normalizable formula w.r.t. $\langle Q,\, x\rangle$ then $Qx.\alpha \overset{*}{\mapsto}_{S_2} Qx.\alpha'$ with α' normalized w.r.t. $\langle Q,\, x\rangle$.*

[3] To this example it is attributed a difficulty of seven points out of ten. In order to make the example easier to follow, we do not translate the formula in negative normal form and suppose that reduce exploits also the rules for the implication. In any case, such rules can be easily derived from those listed in table 1.

Proof. By induction on the number of subformulae in α ($\mathcal{B}(\alpha)$). (To simplify the presentation we consider the case $Q = \forall$). By definition of top normalizable formula, $\alpha = (\beta(x) \vee \gamma)$, with β, γ normalized w.r.t. $\langle \forall, x \rangle$.

($\mathcal{B}(\alpha) = 3$). $\alpha = (P(x) \vee R[x])$ with $P(x)$ and $R[x]$ (distinguished) literals. By rule (4) $\forall x.(P(x) \vee R[x]) \mapsto \forall x.(R[x] \vee P(x))$.

($\mathcal{B}(\alpha) = m+1$). By cases:

- β is minimal w.r.t. $\langle \forall, x \rangle$. γ cannot be minimal w.r.t. $\langle \forall, x \rangle$ (otherwise also α is minimal and hence normalized).
  - (*a*) If $\gamma = \gamma[x]$, by applying rule (4) $\forall x.(\beta(x) \vee \gamma[x]) \mapsto \forall x.(\gamma[x] \vee \beta(x))$.
  - (*b*) If $\gamma = (\eta \wedge \mu)(x)$, by applying rule (8) $\forall x.(\beta(x) \vee (\eta \wedge \mu)(x)) \mapsto \forall x.((\beta(x) \vee \eta) \wedge (\beta(x) \vee \mu))$. Since $max\{\mathcal{B}(\beta(x) \vee \eta), \mathcal{B}(\beta(x) \vee \mu)\} < \mathcal{B}(\beta(x) \vee (\eta \wedge \mu))$ then, by inductive hypothesis, both $(\beta(x) \vee \eta)$ and $(\beta(x) \vee \mu)$ can be normalized. Then $((\beta(x) \vee \eta) \wedge (\beta(x) \vee \mu))$ and hence α are normalizable.
  - (*c*) If $\gamma = (\eta[x] \vee \mu(x))$, by applying rule (7) $\forall x.(\beta(x) \vee (\eta[x] \vee \mu(x))) \mapsto \forall x.(\eta[x] \vee (\beta(x) \vee \mu(x)))$. Since $\mathcal{B}(\beta(x) \vee \mu(x)) < \mathcal{B}(\beta(x) \vee (\eta[x] \vee \mu(x)))$ then, by inductive hypothesis, $(\beta(x) \vee \mu(x))$ is normalizable. Then $(\eta[x] \vee (\beta(x) \vee \mu(x)))$ and hence α are normalizable.

  Notice that this includes also the case in which $\beta = Q'y.\gamma$ since $\beta = Q'y.\gamma$ normalized w.r.t. $\langle Q, x \rangle$ means that it is also minimal w.r.t. $\langle Q, x \rangle$.
- Analogously, if $\beta = (\eta \wedge \mu)(x)$ we can apply rule (6). If $\beta = (\eta[x] \vee \mu(x))$ we can apply rule (5). In both cases, the top normalizable formulae in the resulting formula are normalizable for the induction hypothesis.

Lemma 7. *For any minimal formula α and pair $\langle Q, x \rangle$ $Qx.\alpha \overset{*}{\mapsto}_{S_2} Qx.\alpha'$ with α' normalized w.r.t. $\langle Q, x \rangle$.*

Proof. By induction on the number of subformulae in α ($\mathcal{B}(\alpha)$).

($\mathcal{B}(\alpha) = 1$). α is a literal. Then α is normalized w.r.t. $\langle Q, x \rangle$ for any $\langle Q, x \rangle$.

($\mathcal{B}(\alpha) = m+1$). We know, for the induction hypothesis, that for any minimal formula β such that $\mathcal{B}(\beta) \leq m$, β can be normalized w.r.t. any pair $\langle Q, x \rangle$. By cases:

- $\alpha = (\beta \circ \gamma)$ By inductive hypothesis $Qx.\beta \overset{*}{\mapsto}_{S_2} Qx.\beta'$ and $Qx.\gamma \overset{*}{\mapsto}_{S_2} Qx.\gamma'$ where β', γ' are normalized w.r.t. $\langle Q, x \rangle$. Hence $Qx.(\beta \circ \gamma) \overset{*}{\mapsto}_{S_2} Qx.(\beta' \circ \gamma')$ with $\beta' \circ \gamma'$ normalized w.r.t. $\langle Q, x \rangle$.
- $\alpha = (\beta + \gamma)$. By inductive hypothesis $Qx.\beta \overset{*}{\mapsto}_{S_2} Qx.\beta'$ and $Qx.\gamma \overset{*}{\mapsto}_{S_2} Qx.\gamma'$ where β', γ' are normalized w.r.t. $\langle Q, x \rangle$. Hence $\alpha' = (\beta' + \gamma')$ is either normalized (*e.g.* $\beta' = \beta'[x]$) or a top normalizable formula wrt $\langle Q, x \rangle$. In this last case, $Qx.(\beta' + \gamma')$ can be normalized by lemma 6.
- $\alpha = Q'y.\beta$. Since α is minimal, then it is also normalized wrt any $\langle Q, x \rangle$.

Theorem 8 minimality of $\mathcal{S}$ normal form. *If $\alpha \overset{*}{\mapsto}_{\mathcal{S}} \beta$ then β is minimal.*

Proof. By induction.

(α literal). $\alpha \overset{*}{\mapsto}_S \alpha$ and α is minimal.

($\alpha = (\beta \bullet \gamma)$). With $\bullet \in \{\wedge, \vee\}$. For the induction hypothesis, $\beta \overset{*}{\mapsto}_S \beta'$ and $\gamma \overset{*}{\mapsto}_S \gamma'$ with β' and γ' minimal. Hence $\alpha = (\beta \bullet \gamma) \overset{*}{\mapsto}_S (\beta' \bullet \gamma')$ with $(\beta' \bullet \gamma')$ minimal.

($\alpha = Qx.\beta$). For the induction hypothesis, $\beta \overset{*}{\mapsto}_S \beta'$ with β' minimal. By lemma 7 $\beta' \overset{*}{\mapsto}_{S_2} \beta''$ with β'' normalized w.r.t. (Q, x). By lemma 5 $Qx.\beta'' \overset{*}{\mapsto}_{S_1} \beta'''$ with β''' minimal. Hence $Qx.\beta \overset{*}{\mapsto}_S \beta'''$ with β''' minimal.

We say that a formula α is $\mathcal{S}$-reducible to UE-form (for short $\mathcal{S}$-reducible) if and only if $\alpha \overset{*}{\mapsto}_S \beta$ and β is a UE-formula. Obviously a UE-formula is $\mathcal{S}$-reducible.

The following theorem, while providing a syntactic characterization of a subset of $\mathcal{S}$-reducible formulae, should give evidence of the fact that the set of $\mathcal{S}$-reducible formulae is fairly wide. Let $\forall \mathbf{x}.\phi$ ($\exists \mathbf{y}.\phi$) stands for $\forall x_1 \ldots x_r.\phi$ ($\exists y_1 \ldots y_s.\phi$) for any $r, s \geq 1$.

Theorem 9. *Let $\alpha = \forall \mathbf{y}_n \exists \mathbf{x}_n \ldots \forall \mathbf{y}_i \exists \mathbf{x}_i \ldots \forall \mathbf{y}_1 \exists \mathbf{x}_1.\Phi$. If Φ is a quantifier-free formula such that each literal contains no variables in $\mathbf{y}_k$ and in $\mathbf{x}_l$ with $k < l$, or in $\mathbf{x}_k$ and in $\mathbf{x}_l$ with $k \neq l$, then α is $\mathcal{S}$-reducible to UE-form.*

Proof. By induction on n.

($n = 1$). $\alpha \overset{*}{\mapsto}_S \alpha'$. Since α is a UE-formula, also α' is.

($n = m + 1$). $\alpha = \forall \mathbf{y}_n \exists \mathbf{x}_n.\beta$ where $\beta = \forall \mathbf{y}_m \exists \mathbf{x}_m \ldots \forall \mathbf{y}_1 \exists \mathbf{x}_1.\Phi$. For the induction hypothesis $\beta \overset{*}{\mapsto}_S \beta'$ with β' in UE-form and (by theorem 8) minimal. From minimality it follows that there are no free occurrences of variables in $\mathbf{x}_n$ in the scope of any quantifier in β'. (In Φ and hence also in β' each literal containing $\mathbf{x}_n$ does not contain other bound variables but those (eventually) in $\mathbf{y}_n$). Since β' is in UE-form also $\forall \mathbf{y}_n \exists \mathbf{x}_n.\beta'$ is in UE-form. Hence, by finitely many applications of the rules in $\mathcal{S}$, α can be rewritten into $\forall \mathbf{y}_n \exists \mathbf{x}_n.\beta'$. Finally $\forall \mathbf{y}_n \exists \mathbf{x}_n.\beta' \overset{*}{\mapsto}_S \alpha'$ with α' in UE-form.

As an immediate consequence we have that the monadic class together with two other classes are $\mathcal{S}$-reducible to UE-form and hence decidable.

Corollary 10. *The classes of*

- *monadic formulae,*
- *formulae in which predicates contains at most one bound variable,*
- *formulae in which each predicates either contains no existentially bound variables or, if it contains one, it is the only bound variable it contains,*

are $\mathcal{S}$-reducible to UE-form.

However there are formulae which are $\mathcal{S}$-reducible and are not in the class specified in theorem 9. The formula $\exists x.\forall y.\exists z.(P(x,z) \vee P(y,z))$ (formula number (2) in example 1) is a proof of this fact. On the other hand, a slight variation of a $\mathcal{S}$-reducible formula may not be $\mathcal{S}$-reducible. For example, the formula

$\exists x.\forall y.\exists z.(P(x,z) \wedge P(y,z))$ turns out not to be $\mathcal{S}$-reducible. So far, we have failed to find a simple syntactic characterization of the class of $\mathcal{S}$-reducible formulae.

3 Conclusions and future work

We have proposed a set of rewriting rules which are noetherian, confluent and greatly enlarge the set of formulas which can be proved by a decision procedure for UE-formulas. Example 1 shows also that in some cases the formula result of the reduction process is easier to prove.

We want to emphasize that in this paper we have studied the reducibility to the UE-class given the set $\mathcal{S}$ of rewriting rules. However, the same methodology applies w.r.t. any other (decidable) class and set of rewriting rules. In the future, we plan to extend the above results to other decidable classes maintaining the same set of rewriting rules.

4 Acknowledgements

A first version of the reduction procedure described in this paper was implemented in FOL [2]. Fausto Giunchiglia, Andrea Parodi, Fulvio Rappa and Richard Weyhrauch have provided useful feedback and/or suggestions. This work has been supported by the Italian National Research Council (CNR), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo (Special Project on Information Systems and Parallel Computing).