Available online at www.sciencedirect.com

ScienceDirect

Artificial Intelligence 172 (2008) 1317–1359

Artificial Intelligence

www.elsevier.com/locate/artint

Redundancy in logic III: Non-monotonic reasoning

Paolo Liberatore

Università di Roma "La Sapienza", Dipartimento di Informatica e Sistemistica, Via Ariosto 25, I-00185 Roma, Italy

Received 14 October 2005; received in revised form 15 February 2008; accepted 18 February 2008

Available online 4 March 2008

Abstract

Results about the redundancy of certain versions of circumscription and default logic are presented. In particular, propositional circumscription where all variables are minimized and skeptical default logics are considered. This restricted version of circumscription is shown to have the unitary redundancy property: a CNF formula is redundant (it is equivalent to one of its proper subsets) if and only if it contains a redundant clause (it is equivalent to itself minus one clause); default logic does not have this property in general. We also give the complexity of checking redundancy in the considered formalisms.

Keywords: Logical redundancy; Non-monotonic reasoning; Computational complexity; Circumscription; Default logic

1. Introduction

In this paper, we study the problem whether a circumscriptive [36] or default [44] theory is redundant, that is, it contains unnecessary parts. Formally, a theory is redundant if it is equivalent to one of its proper subsets; a part is redundant in a theory if the theory is not semantically changed by the removal of the part. The problem of redundancy in other settings has been extensively analyzed in the literature. The complexity of establishing whether a CNF, 2CNF, and Horn formula is redundant has been studied by the author of the present article [26,30], who also analyzed some problems related to irredundant equivalent subsets of formulae. These problems are all considered in the settings of classical propositional logic.

The related problem of minimizing a propositional theory, in particular when in Horn form, has been analyzed by several authors [1,22,23,34,37,52]. Minimization is significantly different from redundancy: a formula is minimal if there is no shorter formula that is equivalent to it. Minimal formulae are also irredundant, but not the other way around: a formula may be irredundant because no part can be removed from it, but still a completely different formula is both shorter than it and equivalent to it. For example, $\{a \vee b, a \vee \neg b\}$ is irredundant (because none of the two clauses can be removed from it while maintaining equivalence with the original set) but is not minimal (because it is equivalent to the shorter set $\{a\}$). There are motivations for studying both minimization and redundancy. Minimization produces the shortest possible formula equivalent to a given one, and this has obvious advantages; making a formula irredundant may not produce a formula as short as the minimal ones, but has the additional advantage of not changing the structure of the original formula, but only to remove parts from it.

E-mail address: paolo@liberatore.org.

0004-3702/$ – see front matter
doi:10.1016/j.artint.2008.02.003

A related problem is that of studying the properties of formulae that are already known to be irredundant. Büning and Zhao [7] studied the problems of equivalence and extension-equivalence of irredundant formulae. Also related is the problem of minimal unsatisfiability, that is, checking whether an unsatisfiable formula would become satisfiable as soon as a clause is removed from it [5,14,43].

Other authors have studied redundancy in settings different from that in this article. Ginsberg [18] and Schmolze and Snyder [48] studied the redundancy of production rules. Gottlob and Fermüller [17] studied the redundancy of a literal within a clause in first-order logic.

The settings considered in this article are those of circumscription and default logic, which are two of the most studied [2,4,6,10,13,29,31,36,38,44] forms of non-monotonic reasoning, as opposite to classical logic, which is monotonic. A logic is monotonic if the consequences of a set of formulae monotonically non-decrease with the set. In other words, all formulae that are entailed by a set are also entailed by every superset of it. Circumscription and default logic do not have this property, and are therefore non-monotonic. For circumscription, we assume that all variables are minimized; the rationale for this restriction is that fixed and varying variables can be efficiently eliminated [8,9], which shows that the minimized variables are the "core" of the circumscription formalism. Other authors have indeed considered circumscription only under this restriction [6,25,40]. This shows that this restriction of circumscription is of interest; however, results about redundancy in this case do not necessarily extend to the general case, as discussed in the conclusions.

For default logic, there are several semantics; for most of them, one can choose between the "credulous" and "skeptical" approach. In this article, we consider the skeptical approach under the original semantics [44] and three similar ones: justified [33], constrained [11,46], and rational [39]. We however also consider the case in which we assume that the semantics of a theory is the set of its extensions, without combining these extensions in a skeptical manner. The results obtained in this case hold for the credulous approach under the original semantics (where not otherwise stated, the skeptical approach is assumed). The properties of redundancy in the credulous approach under the other semantics is left open. Since redundancy is defined in terms of equivalence (namely, equivalence of a formula to a proper subset of it), it is affected by the kind of equivalence used. In particular, equivalence can be defined in two ways for default logic: equality of extensions and equality of consequences. This leads to two different definitions of redundancy in default logic.

Both circumscription and default logic differ from classical propositional logic. This difference affects redundancy. If a CNF formula contains a redundant clause, it is redundant (equivalent to one of its proper subsets); the converse is true in propositional logic, but not in all logics. In particular, it may be the case that a formula is equivalent to one of its proper subsets, but none of its clauses is redundant. We will indeed show a situation in default logic where $\{a, b\}$ is equivalent to $\emptyset$ but neither to $\{a\}$ nor to $\{b\}$, which means that $\{a, b\}$ is redundant but does not contain any single redundant element.

The property that a formula is redundant if and only if it contains a redundant clause is called *unitary redundancy*. Classical logic has this property; other logics, like default logic, do not. We show three different sufficient conditions for this property to hold in a logic; one of them involves monotonicity, another involves cumulativity [35]. A property that entails unitary redundancy is that of monotonic redundancy: if $\Pi' \subseteq \Pi'' \subseteq \Pi$ and Π' and Π are equivalent then Π'' and Π are equivalent as well. This is the property for which the sufficient conditions are actually proved; unitary redundancy follows.

Regarding the specific non-monotonic formalisms considered here, we show that monotonic redundancy, and therefore unitary redundancy, holds for circumscription and for the redundancy of the background theory in default logic when all defaults are categorical (prerequisite-free) and normal. In the general case, default logic does not have the unitary redundancy property (and therefore does not have the monotonic redundancy property either). We also considered the redundancy of defaults in a default theory. In this case, monotonic and unitary redundancy hold for justified default logic but not for the other three considered semantics.

Regarding the complexity results, we show that checking whether a clause is redundant in a formula and whether a formula is redundant according to circumscriptive inference are Π_2^p-complete problems. For default logic, the results are as follows. Checking redundancy, based on extensions, of a clause in the background theory is Π_2^p-complete for Reiter and justified default logics, and Π_3^p-complete for constrained and rational default logic; checking redundancy based on skeptical consequences is Π_3^p-complete for all four semantics. Checking redundancy of a background theory is Σ_3^p-complete and Σ_4^p-complete, respectively, for equivalence based on extensions and skeptical consequences. The proofs of the latter two results are of some interest by themselves, as they are done by first showing that the problems are Π_2^p-complete and Π_3^p-complete, respectively, and then showing that such complexity results can be raised one level in the polynomial hierarchy. This technique allows for a proof of hardness for a class such as Σ_4^p without involving complicated QBFs such as $\exists W \forall X \exists Y \forall Z.F$. Regarding the credulous approach, we prove that equality of extensions and equality of credulous consequences coincide for the Reiter default logic, but not for the other three considered semantics. Regarding the redundancy of defaults, we only considered Reiter default logic with equivalence based on equality of extensions; we proved that the redundancy of a default is Π_2^p-complete and the redundancy of a set of defaults is Σ_3^p-complete in this case.

For the sake of clarity, long proofs and very technical parts are in the appendix.

2. Preliminaries

If Π and Γ are sets, $\Pi \backslash \Gamma$ denotes the set of elements that are in Π but not in Γ. This operator is often called set subtraction because the elements of Γ are "subtracted" from Π. An alternative definition of this operator is: $\Pi \backslash \Gamma = \Pi \cap \overline{\Gamma}$, where $\overline{\Gamma}$ is the complement of Γ.

All formulae considered in this paper are propositional and finite Boolean formulae over a finite alphabet. We typically use formulae in CNF, that is, sets of clauses; we refer to sets of clauses simply as formulae. $Var(\Pi)$ is the set of variables mentioned in the formula Π, $Cn(\Pi)$ is the set of its consequences (formulae entailed by it). If S a (possibly infinite) set of formulae (all built of variables from the same finite alphabet), $\vee S$ denotes a formula whose models are exactly the models that satisfy all formulae of S.

We use the notation $l \in \gamma$, where l is a literal and γ a clause, to indicate that l is a literal of the clause γ. In some places, we use the notation $\neg\gamma$, where γ is a clause, to denote the formula $\{\neg l \mid l \in \gamma\}$. Note that γ is a clause, while both $\{\gamma\}$ and $\neg\gamma$ are formulae (sets of clauses). A clause is positive if and only if it contains only positive literals.

A propositional interpretation is an assignment from a set of propositional variables to the set $\{\mathsf{true}, \mathsf{false}\}$. We denote a model by the set of variables it assigns to true. We use the notation $Mod(\Pi)$ to denote the set of models of a formula Π. We sometimes use models as formulae, e.g., $\Pi \wedge \omega$ where Π is a formula and ω is a model. In the context where a formula is expected, a model ω represents the formula $\{x \mid x \in \omega\} \cup \{\neg x \mid x \notin \omega\}$. If Π is a formula and ω_X is an interpretation over the set of variables X, we denote by $\Pi|_{\omega_X}$ the formula obtained by replacing each variable of X with its value as assigned by ω_X in Π. Since a model is a set of positive literals, $M \subseteq M'$ holds if and only if M assigns to false all variables that M' assigns to false.

All formulae considered in this article are assumed to be built over a finite alphabet of variables. Sometimes, formulae are built from other formulae with the addition of new variables. These additions implicitly assume the existence of a countable set of new variables to take from; these new variables are symbols not appearing in the original formulae.

3. Arbitrary logics

In this section, we show some results about redundancy in an arbitrary logic. To this aim, we consider a logic to be characterized by an entailment relation $\models_L$ and an equivalence relation $\equiv_L$. In the basic case, these relations will be over the set of all sets of clauses over a given alphabet, so that one can write $\Pi \equiv_L \Pi'$ if Π and Π' are sets of clauses. However, we also analyze the question of redundancy of defaults, which depends on whether two sets of defaults D and D' are equivalent, that is, $D \equiv_L D'$. In order to obtain truly general results, in this section we do not make assumptions over the basic elements in these sets, and only assume $\models_L$ and $\equiv_L$ to be relations over the sets of elements from a given set; in this article, this set is either the set of all clauses or of all default rules over a given alphabet, but the results in this section hold for an arbitrary set.

We do not implicitly assume any property over these two relations or over their relationship; as it will be shown, some entailment and equivalence relations we consider do not have the properties one would generally expect from them; for example, a definition of equivalence that is instrumental to the study of default logic is not transitive. Formally, that would mean that this form of logical equivalence is not in fact an equivalence relation. While the term "equivalence" could be considered misleading, it is still the most expressive name for this relation.

Some properties entailment and equivalence relations may have are defined as follows.

Definition 1. The following properties are defined over relations $\models_L$, $\equiv_L$, where Π, Π', and Π'' are sets:

Reflexivity: $\Pi \models_L \Pi$, $\Pi \equiv_L \Pi$;
Symmetry: if $\Pi \equiv_L \Pi'$ then $\Pi' \equiv_L \Pi$;
Transitivity: if $\Pi \equiv_L \Pi'$ and $\Pi' \equiv_L \Pi''$ then $\Pi \equiv_L \Pi''$;
Right weakening: if $\Pi \models_L \Pi'$ and $\Pi'' \subseteq \Pi'$ then $\Pi \models_L \Pi''$;
Right conjunction: if $\Pi \models_L \Pi'$ and $\Pi \models_L \Pi''$ then $\Pi \models_L \Pi' \cup \Pi''$;
Monotonicity: if $\Pi \models_L \Pi'$ and $\Pi \subseteq \Pi''$ then $\Pi'' \models_L \Pi'$
Cumulativity: if $\Pi \models_L \Pi'$ then $\Pi \equiv_L \Pi \cup \Pi'$;
Half-entailment: if $\Pi \equiv_L \Pi'$ then $\Pi \models_L \Pi'$;
Full-entailment: $\Pi \equiv_L \Pi'$ if and only if $\Pi \models_L \Pi'$ and $\Pi' \models_L \Pi$.

Reflexivity, symmetry, and transitivity are defined as they usually are on binary relations. Some other of these conditions are defined differently from the standard way because we cannot use propositional connectives and entailment. This is due to the intentional lack of assumptions over the elements in the sets Π, Π', etc: since these are not necessarily sets of clauses or of propositional formulae, one cannot write $\Pi' \models \Pi''$ where $\models$ is propositional entailment. As a result, one cannot write for example right weakening in the usual form: if $\Pi \models_L \Pi'$ and $\Pi' \models \Pi''$ then $\Pi \models_L \Pi''$. The above definitions, being based on set operations only, can be used in scenarios where propositional connectives and entailment are not defined, such as the redundancy of defaults (if D and D' are sets of defaults, $D \models D'$ is undefined if $\models$ is propositional entailment).

The above definitions are equivalent to the classical ones for sets of clauses or propositional formulae with the additional assumption that $\models_L$ and $\equiv_L$ behave the same on propositionally equivalent formulae: if Π and Π' are propositionally equivalent then for every Π'' it holds $\Pi \models_L \Pi''$ if and only if $\Pi' \models_L \Pi''$ holds, etc. In this case, the definitions above coincide with the classical ones by taking sets that are deductively closed.

All entailment and equivalence relations considered in this article are reflexive and all equivalence relations are also symmetric. However, a definition of equivalence used for the study of default logic will be shown not transitive, and some forms of entailment will be shown not to enjoy the right weakening property. Equivalence will often be defined according to the full entailment property from an entailment relation; in such cases, full entailment holds by definition.

An element is redundant in a set if its removal leads to an equivalent set.

Definition 2 *(Redundancy of an element).* An element γ of a set Π is redundant in it according to $\equiv_L$ if $\Pi \equiv_L \Pi \backslash \{\gamma\}$.

A set is redundant if it is equivalent to one of its proper subsets.

Definition 3 *(Redundancy of a set).* A set Π is redundant according to $\equiv_L$ if there exists $\Pi' \subset \Pi$ such that $\Pi \equiv_L \Pi'$.

In propositional logic, these two forms of redundancy are related: a set of clauses is redundant if and only if it contains a redundant clause. The same holds for circumscription but not for all logics. For example, it does not hold for the redundancy of clauses in the background theory of default logic.

Definition 4 *(Unitary redundancy).* Equivalence $\equiv_L$ has the unitary redundancy property if the redundancy of a set is the same as the presence of a redundant element in it.

More formally, if $\equiv_L$ has this property then $\Pi \equiv_L \Pi'$ and $\Pi' \subset \Pi$ imply that there exists $\gamma \in \Pi$ such that $\Pi \equiv_L \Pi \backslash \{\gamma\}$. One may also assume that such γ is in $\Pi \backslash \Pi'$; modifying the definition in this way would not change the results in this article.

A property of $\equiv_L$ that implies unitary redundancy is the following limited form of monotonicity.

Definition 5 *(Monotonic redundancy).* Equivalence $\equiv_L$ has the monotonic redundancy property if, for every three sets Π, Π', and Π'' such that $\Pi \equiv_L \Pi'$ and $\Pi' \subseteq \Pi'' \subseteq \Pi$, it holds that $\Pi'' \equiv_L \Pi$.

Monotonic redundancy implies that every set that is contained in another and contains yet another is equivalent to the former if the former and the latter are equivalent to each other. In this definition, $\Pi'' \equiv_L \Pi$ cannot in general be replaced with $\Pi'' \equiv_l \Pi'$ unless $\equiv_L$ is transitive.

Monotonic redundancy implies unitary redundancy: if Π is equivalent to Π' with $\Pi' \subset \Pi$, then for an arbitrary $\gamma \in \Pi \backslash \Pi'$ it holds that $\Pi \backslash \{\gamma\}$ is equivalent to Π. Whenever possible, monotonic redundancy will be proved. Since counterexamples of the unitary redundancy property are stronger than monotonic redundancy ones, unitary redundancy is used in this article when giving counterexamples.

While monotonic redundancy entails unitary redundancy, the converse does not hold.

Counterexample 1. *There exists a logic where $\models_L$ is reflexive, $\equiv_L$ is symmetric and transitive, they are related by full-entailment, and unitary redundance holds, but monotonic redundancy does not.*

The following three theorems provide some sufficient conditions to monotonic redundancy, and therefore to unitary redundancy as well.

Theorem 1. *If $\models_L$ is reflexive and enjoys right weakening, $\equiv_L$ is transitive, and they are related by full-entailment, then $\equiv_L$ has the monotonic redundancy property.*

Proof. Let us assume $\Pi \equiv_L \Pi'$ and $\Pi' \subseteq \Pi'' \subseteq \Pi$, and prove $\Pi \equiv_L \Pi''$. Since $\Pi \equiv_L \Pi'$, by full-entailment we have $\Pi' \models_L \Pi$. Since $\Pi'' \subseteq \Pi$, by right weakening it holds $\Pi' \models_L \Pi''$. Since $\models_L$ is reflexive, it also holds $\Pi'' \models_L \Pi''$. By right weakening, $\Pi'' \models_L \Pi'$. Since we previously proved that $\Pi' \models_L \Pi''$, by full-entailment it follows $\Pi' \equiv_L \Pi''$. Since $\Pi \equiv_L \Pi'$, by symmetry (which is a consequence of full-entailment) and transitivity of $\equiv_L$ it follows that $\Pi \equiv_L \Pi''$. $\square$

Theorem 2. *If $\models_L$ is reflexive and monotonic and is related to $\equiv_L$ by full entailment, then $\equiv_L$ has the monotonic redundancy property.*

Proof. Let us assume that $\Pi \equiv_L \Pi'$ and that $\Pi' \subseteq \Pi'' \subseteq \Pi$. By full entailment, the first condition implies $\Pi' \models_L \Pi$. Since $\Pi' \subseteq \Pi''$, by monotonicity we have $\Pi'' \models_L \Pi$. By reflexivity of $\models_L$, we have $\Pi'' \models_L \Pi''$; since $\Pi'' \subseteq \Pi$, by monotonicity we have that $\Pi \models_L \Pi''$. Since we have already proved that $\Pi'' \models_L \Pi$ holds, we can use full entailment and conclude that $\Pi'' \equiv_L \Pi$. $\square$

Theorem 3. *If $\models_L$ enjoys right weakening, $\equiv_L$ is symmetric and transitive, and they are related by cumulativity and half-entailment, then $\equiv_L$ has the monotonic redundancy property.*

Proof. Assume that $\Pi \equiv_L \Pi'$ and $\Pi' \subseteq \Pi'' \subseteq \Pi$. By symmetry and half-entailment, $\Pi' \models_L \Pi$. By right weakening, we have $\Pi' \models_L \Pi'' \backslash \Pi'$. By cumulativity, $\Pi' \equiv_L (\Pi'' \backslash \Pi') \cup \Pi''$, which is the same as $\Pi' \equiv_L \Pi''$. By transitivity and symmetry of $\equiv_L$, we have $\Pi \equiv_L \Pi''$. $\square$

Another general result is that redundancy of an element is the same as the rest of the set entailing that element provided that the logic has some properties. The existence of this correspondence has been suggested by one of the referees.

Theorem 4. *If $\models_L$ and $\equiv_L$ have the half-entailment and cumulativity properties and $\models_L$ has the right weakening property then an element γ is redundant in a set Π if and only if $\Pi \backslash \{\gamma\} \models_L \gamma$.*

Proof. Let us assume that $\Pi \backslash \{\gamma\} \equiv_L \Pi$ and $\gamma \in \Pi$. By half-entailment, we have $\Pi \backslash \{\gamma\} \models_L \Pi$. By right weakening, $\Pi \backslash \{\gamma\} \models_L \gamma$.

To prove the opposite direction, assume $\Pi \backslash \{\gamma\} \models_L \gamma$. By cumulativity, it holds $\Pi \backslash \{\gamma\} \equiv_L \Pi \backslash \{\gamma\} \cup \{\gamma\}$, which is the same as $\Pi \backslash \{\gamma\} \equiv_L \Pi$. $\square$

The same equivalence can also be proved using different conditions.

Theorem 5. *If $\models_L$ satisfies reflexivity, right weakening, right conjunction, and is related to $\equiv_L$ by full entailment, then an element γ is redundant in a set Π if and only if $\Pi \backslash \{\gamma\} \models_L \gamma$.*

Proof. If $\Pi \backslash \{\gamma\} \equiv_L \Pi$, then $\Pi \backslash \{\gamma\} \models_L \Pi$ by full entailment, and therefore $\Pi \backslash \{\gamma\} \models_L \gamma$ by right weakening. If $\Pi \backslash \{\gamma\} \models_L \gamma$, we can apply right conjunction together with $\Pi \backslash \{\gamma\} \models_L \Pi \backslash \{\gamma\}$, which holds by reflexivity, to obtain $\Pi \backslash \{\gamma\} \models_L \Pi$. Since $\Pi \models_L \Pi \backslash \{\gamma\}$ holds by reflexivity and right weakening, by full entailment we have $\Pi \backslash \{\gamma\} \equiv_L \Pi$. $\square$

While $\Pi \backslash \{\gamma\} \models_L \gamma$ could be used as an alternative definition of redundancy of γ, its equivalence to $\Pi \backslash \{\gamma\} \equiv_L \Pi$ does not hold for every logic, as shown by the following counterexample.

Counterexample 2. *There exists two relations $\models_L$ and $\equiv_L$ such that $\models_L$ is reflexive, $\equiv_L$ is reflexive and symmetric, $\models_L$ and $\equiv_L$ are related by full entailment, but there exists Π and $\gamma \in \Pi$ such that $\Pi \backslash \{\gamma\} \models_L \gamma$ but $\Pi \not\equiv_L \Pi \backslash \{\gamma\}$.*

A natural question is which definition of redundancy would be the most natural for logics where they differ. If "redundancy" is taken as its literal meaning of "something is unnecessary", then a clause would be redundant if it can be taken out of a set without this removal changing the meaning of the set. Formally, $\Pi \backslash \{\gamma\} \equiv_L \Pi$.

4. Circumscription

Circumscriptive inference is based on the minimal models of a theory, i.e., the models whose set of variables assigned to true is minimal w.r.t. set inclusion. Formally, we define the set of minimal models as follows.

Definition 6. The set of minimal models of a propositional formula Π, denoted by $\mathrm{CIRC}(\Pi)$, is defined as follows.

$$\mathrm{CIRC}(\Pi) = \min_{\subseteq}\big(Mod(\Pi)\big)$$

We define $\mathrm{CIRC}(\Pi)$ to be a set of models instead of a formula, although the latter is more common in the literature. Circumscriptive entailment is defined like classical entailment but only minimal models are taken into account.

Definition 7. The circumscriptive inference $\models_M$ is defined by: for any two formulae Π and Γ, it holds $\Pi \models_M \Gamma$ if Γ is satisfied by all minimal models of Π:

$$\Pi \models_M \Gamma \quad \text{if } \mathrm{CIRC}(\Pi) \subseteq Mod(\Gamma)$$

We assume that the alphabet is finite. This definition is actually the restriction of the original circumscription [36] without "fixed" and "varying" variables. We use this version because it captures the core of circumscription, that of using minimal models, as shown by the translations that efficiently remove fixed [9] and varying [8] variables. The importance of this subcase is also shown by the attemption it has been given in the literature [6,25,40]. We however remark that, while an interesting subcase, the results about redundancy given in this article do not necessarily extend to the general case, as the elimination of fixed and varying predicates modify the formula.

Equivalence in propositional logic can be defined in two equivalent ways: either by equality of the models or by equality of the sets of entailed formulae. These two definitions of equivalence coincide for circumscriptive inference as well. We define $\equiv_M$ as follows: $\Pi \equiv_M \Gamma$ if and only if $\mathrm{CIRC}(\Pi) = \mathrm{CIRC}(\Gamma)$. The next theorem gives some properties of circumscriptive inference and equivalence.

Theorem 6. *Relation $\models_M$ has the properties of reflexivity and right weakening; relation $\equiv_M$ is reflexive, symmetric, and transitive; these two relations are related by cumulativity and full-entailment.*

Proof. Reflexivity, symmetry, transitivity, and right weakening are obvious from definition. Cumulativity of circumscription is well-known [35,47] when equivalence is defined as equality of consequences, which is the same as $\equiv_M$ by definition of $\models_M$ and $\equiv_M$.

We now prove that $\models_M$ and $\equiv_M$ are related by full-entailment. If $\Pi \equiv_M \Pi'$, then the minimal models of Π and Π' are the same. Therefore, the minimal models of Π are also all models of Π', which means that $\Pi \models_M \Pi'$.

For the other direction, let us assume that $\Pi \models_M \Pi'$ and $\Pi' \models_M \Pi$, and prove $\Pi \equiv_M \Pi'$. To the contrary, assume that Π has a minimal model M which is not a minimal model of Π'. Since M is a minimal model of Π and $\Pi \models_M \Pi'$, it follows that M is a model of Π'. Since it is a model but not a minimal model, there exists $M' \subset M$ that is a minimal model of Π'. However, since $\Pi' \models_M \Pi$, this model M' is also a model of Π. This contradicts the assumption that M is a minimal model of Π. In a similar way, assuming that Π' has a minimal model which is not a minimal model of Π leads to a contradiction. □

Redundancy of a clause is defined as follows.

Definition 8. A clause $\gamma \in \Pi$ is CIRC-redundant in formula Π if and only if $\Pi \backslash \{\gamma\} \equiv_M \Pi$.

By Theorem 4 and Theorem 6, redundancy of a clause γ in a set Π could also be defined as $\Pi \backslash \{\gamma\} \models_M \gamma$. A formula is redundant if some of its clauses can be removed without changing its semantics.

Definition 9. A formula is CIRC-redundant if it is $\equiv_M$-equivalent to one of its proper subsets.

A formula is therefore redundant if some clauses can be removed from it while preserving equivalence. Monotonic redundancy can be easily proved to hold given the properties of $\models_M$ and $\equiv_M$.

Theorem 7. *Circumscription has the monotonic redundancy property.*

Proof. Theorem 6 proves that $\models_M$ and $\equiv_M$ have all properties that are necessary to apply Theorem 1. □

As a result, circumscription also has the unitary redundancy property.

Corollary 1. *Circumscription has the unitary redundancy property.*

4.1. Redundant clauses

The following lemma characterizes the clauses that are redundant in a formula.

Lemma 1. *The following three conditions are equivalent for every formula Π and clause $\gamma \in \Pi$.*

1. *γ is* CIRC*-redundant in Π;*
2. *for each $M \in Mod(\Pi \backslash \{\gamma\} \cup \neg\gamma)$ there exists $M' \in Mod(\Pi)$ such that $M' \subset M$;*
3. *for each $M \in Mod(\Pi \backslash \{\gamma\} \cup \neg\gamma)$ there exists $M' \in Mod(\Pi \backslash \{\gamma\})$ such that $M' \subset M$.*

Proof. The models of $\Pi \backslash \{\gamma\}$ that are not models of Π are exactly the models of $\Pi \backslash \{\gamma\} \cup \neg\gamma$. The two formulae Π and $\Pi \backslash \{\gamma\}$ are $\models_M$-equivalent if none of these models (if any) is minimal, that is, all these models contain other models of Π. In other words, γ is redundant if and only if every model of $\Pi \backslash \{\gamma\} \cup \neg\gamma$ properly contains a model of Π.

The fact that we can check $M' \in Mod(\Pi \backslash \{\gamma\})$ instead of $M' \in Mod(\Pi)$ follows from the fact that $Mod(\Pi \backslash \{\gamma\})$ is composed of all models of Π and all models of $\Pi \backslash \{\gamma\} \cup \neg\gamma$. Consider a model M that is a minimal model of $\Pi \backslash \{\gamma\} \cup \neg\gamma$. The condition $M' \subset M$ implies that M' is not a model of $\Pi \backslash \{\gamma\} \cup \neg\gamma$, and is therefore a model of Π. By transitivity, the condition that there exists $M' \in Mod(\Pi)$ such that $M' \subset M$ holds for all models of $\Pi \backslash \{\gamma\} \cup \neg\gamma$. □

Computationally, checking the second or third condition of this lemma can be done by checking whether for all $M \in \ldots$ there exists $M' \in \ldots$ such that a simple condition is met. As a result, the problem is in Π_2^p. For positive clauses, checking CIRC-redundancy is easier, as it amounts to checking classical redundancy.

Lemma 2. *A positive clause is* CIRC*-redundant in a formula if and only if it is classically redundant in the formula.*

Proof. Theorem 4 and Theorem 6 prove that γ is CIRC-redundant in Π if and only if $\Pi\backslash\{\gamma\} \models_M \gamma$. Bossu and Siegel [6, Property 1.6.6] proved that a positive clause is sub-implied by a formula if and only if it is (classically) implied by it. This implies the statement of the lemma because sub-implication coincides with our version of circumscription in the propositional case; this is also essentially proved by Bossu and Siegel [6, Property 1.6.3], considering that all models are discriminant in the propositional case. □

Intuitively, a positive clause only excludes models that assign false to all variables in the clause. Therefore, whenever a positive clause is classically irredundant, it is because such models were not otherwise excluded; therefore, it is also CIRC-irredundant.

According to this argument, it may look like all negative clauses are CIRC-redundant, since they exclude models with positive literals and these models are not minimal. This is however not the case: a model with some positive literals might be minimal because no other model of the formula has a smaller (w.r.t. set inclusion) set of variables assigned to true. Consider, for example, the following formula:

$$\Pi = \{\neg x_1 \vee \neg x_2, x_1 \vee x_3, x_2 \vee x_3\}$$

The clause $\neg x_1 \vee \neg x_2$, although negative, is CIRC-irredundant. Indeed, $\Pi\backslash\{\neg x_1 \vee \neg x_2\} = \{x_1 \vee x_3, x_2 \vee x_3\}$, and this formula has $\{x_1, x_2\}$ and $\{x_3\}$ as its minimal models. The first one is not a model of Π because of the clause $\neg x_1 \vee \neg x_2$. Therefore, $\neg x_1 \vee \neg x_2$ is CIRC-irredundant in Π.

Intuitively, a negative clause excludes the possibility of setting all variables to true, while minimal inference only *tries* to set variables to false. Therefore, removing the clause may generate a model that have its variables set to true ($\{x_1, x_2\}$ in the example), but is minimal because of the values of the other variables (x_3 in the example).

Lemma 2 can be extended to clauses containing negative literals via the addition of new clauses and new variables.

Lemma 3. *A clause γ is classically redundant in a formula Π if and only if it is* CIRC*-redundant in $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}$, where each variable x' is a new variable associated to each variable x such that $\neg x \in \gamma$.*

Proof. In propositional logic, γ is redundant in Π if and only if $\Pi\backslash\{\gamma\}$ entails γ. If γ is redundant in Π then $\Pi\backslash\{\gamma\} \models \gamma$ and therefore $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}\backslash\{\gamma\} \models \gamma$. Since γ is redundant in $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}$, it is also CIRC-redundant.

Let us now assume that γ is irredundant in Π, that is, $\Pi\backslash\{\gamma\} \cup \neg\gamma$ has some models. Let M be one such model. Since this model satisfies $\neg\gamma$, it assigns false to any variable x such that $x \in \gamma$ and true to any variable x such that $\neg x \in \gamma$. Extending M to assign false to all variables x', this model also satisfies $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}\backslash\{\gamma\} \cup \neg\gamma$.

We show that M does not contain a model of $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}$; by Lemma 1, this implies that γ is not redundant in $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}$. This extended model M assigns false to all $x \in \gamma$ and also false to all x' such that $\neg x \in \gamma$. On the other hand, $\{\gamma\} \cup \{x \vee x' \mid \neg x \in \gamma\}$ entails the clause $\bigvee\{x \mid x \in \gamma\} \vee \bigvee\{x' \mid \neg x \in \gamma\}$; this can be proved for example by iteratively resolving upon all literals x such that $\neg x \in \gamma$. As a result, no model of $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}$ has a model that assign false to all $x \in \gamma$ and all x' such that $\neg x \in \gamma$. Since this is instead done by M, it follows that no model of $\Pi \cup \{x \vee x' \mid \neg x \in \gamma\}$ is contained in M. □

Note that the clauses $x \vee x'$ are not necessarily CIRC-irredundant in the considered formula. On the other hand, Lemma 2 can be applied to them: they are CIRC-redundant if and only if they are classically redundant.

4.2. Complexity of CIRC-redundancy

We first show the complexity of checking the redundancy of a single clause.

Theorem 8. *Checking the CIRC-redundancy of a clause in a formula is Π_2^p-complete, and remains hard if the formula is assumed to be satisfiable.*

In order to characterize the complexity of the problem of checking the CIRC-redundancy of a formula, we use the fact that a formula is CIRC-redundant if and only if it contains a CIRC-redundant clause by Corollary 1. In particular, Theorem 8 shows that the problem of checking the CIRC-redundancy of a clause γ in Π is Π_2^p-hard. In order for this result to be used as a proof of hardness for the problem of CIRC-redundancy of formulae, we need to modify the formula Π in such a way all its clauses but γ are made CIRC-irredundant. This is the corresponding of Lemma 4 of the article about redundancy of propositional CNF formulae [26], which has been useful because it allows to "localize" problems about redundancy. In particular, for every consistent formula and arbitrary subset of it, we can change the formula in such a way the clauses not in the subset are made irredundant.

Definition 10. For every formula $\Pi = \{\gamma_1, \ldots, \gamma_m\}$ over variables $X = \{x_1, \ldots, x_n\}$ and every $\Pi' \subseteq \Pi$, the formula $I(\Pi, \Pi')$ is defined as follows, where a, b, s, and t are new variables and $C = \{c_i \mid 1 \leqslant i \leqslant m\}$, $D = \{d_i \mid 1 \leqslant i \leqslant m\}$, and $X' = \{x'_i \mid 1 \leqslant i \leqslant n\}$ are sets of new variables.

$$\begin{aligned}
I(\Pi, \Pi') = \{s \vee t\} \cup \{s \vee a, t \vee b\} \cup \\
&\{\neg s \vee t \vee c_i \vee d_i \mid 1 \leqslant i \leqslant m\} \cup \{\neg s \vee \neg c_i \mid 1 \leqslant i \leqslant m\} \cup \\
&\{\neg t \vee c_i \vee \gamma_i \mid 1 \leqslant i \leqslant m,\ \gamma_i \in \Pi \backslash \Pi'\} \cup \\
&\{s \vee \neg t \vee x_i \vee x'_i \mid 1 \leqslant i \leqslant n\} \cup \\
&\{\neg s \vee \neg t \vee \gamma_i \mid 1 \leqslant i \leqslant m,\ \gamma_i \in \Pi'\}
\end{aligned}$$

The following lemma shows that the clauses of $I(\Pi, \Pi')$ that do not originate from clauses of Π' are irredundant, and that the clauses that originate from clauses of Π' maintain their redundancy status.

Lemma 4. *For every consistent formula $\Pi = \{\gamma_1, \ldots, \gamma_m\}$ and every $\Pi' \subseteq \Pi$, the only* CIRC*-redundant clauses of $I(\Pi, \Pi')$ are the clauses $\neg s \vee \neg t \vee \gamma_i$ such that $\gamma_i \in \Pi'$ is* CIRC*-redundant in Π.*

This lemma shows a way to make the clauses of $\Pi \backslash \Pi'$ *necessary*, that is, contained in every equivalent subset of Π. This theorem allows to characterize the complexity of formula CIRC-redundancy.

Theorem 9. *The problem of verifying* CIRC*-redundancy is Π_2^p-complete.*

Proof. By Corollary 1, Π is redundant if and only if it contains a redundant clause. Therefore, we have to solve a linear number of problems in Π_2^p. Since these problems can be solved in parallel, the whole problem is in Π_2^p.

Hardness is proved by reduction from the problem of CIRC-redundancy of a single clause. By Lemma 4, a clause γ is CIRC-redundant in a consistent formula Π if and only if $\neg s \vee \neg t \vee \gamma$ is CIRC-redundant in $I(\Pi, \{\gamma\})$ and all other clauses of $I(\Pi, \{\gamma\})$ are irredundant. Since the problem of checking redundancy of a clause in a formula is Π_2^p-hard even if the formula is satisfiable by Theorem 8, the problem of formula redundancy is Π_2^p-hard as well. □

5. Default logic

A default theory is a pair $\langle D, W \rangle$, where W is formula and D is a set of default rules, each rule being in the form:

$$\frac{\alpha : \beta}{\gamma}$$

The formulae α, β, and γ are called the precondition, the justification, and the consequence of the default, respectively. In this paper, we assume that W is a finite CNF formula (a finite set of clauses) and that the set of variables and defaults are finite. A variant we do not consider is that in which a set of formulae $\beta_1, \ldots, \beta_m$ is in place of the single formula β. Given a default $d = \frac{\alpha:\beta}{\gamma}$, its parts are denoted by $\mathrm{prec}(d) = \alpha$, $\mathrm{just}(d) = \beta$, and $\mathrm{cons}(d) = \gamma$.

A default d is normal if $\mathrm{just}(d) = \mathrm{cons}(d)$; it is categorical if $\mathrm{prec}(d) = \mathsf{true}$. Defaults that are normal and possibly categorical have been often considered because of their semantical properties and their ability to encode natural domains [4,41]. In particular, circumscription with all variables minimized can be easily simulated by defaults in the form $\frac{:\neg x}{\neg x}$.

We use the operational semantics of default logics [2,3,15,16], which is based on sequences of defaults with no duplicates. If Π is such a sequence, we denote by $\Pi[d]$ the sequence of defaults preceding d in Π, and by $\Pi \cdot [d]$ the sequence obtained by adding d at the end of Π. We extend the notation from defaults to sequences, so that $\mathrm{prec}(\Pi)$ is the conjunction of all preconditions of the defaults in Π, $\mathrm{just}(\Pi)$ is the conjunction of all justifications, and $\mathrm{cons}(\Pi)$ is the conjunction of all consequences.

Propositional entailment is denoted by $\models$, $\top$ indicates (combined) consistency, and $\perp$ indicates inconsistency. For example, $A \top B$ means that $A \wedge B$ is consistent, while $A \perp B$ means that $A \wedge B$ is inconsistent.

Default logic can be defined in terms of the *selected processes*, that are the sequences of defaults that are considered applicable by the considered semantics [2]. These are defined as a combinations of conditions.

Definition 11 *(Process).* A sequence of defaults Π containing no duplicate is a process if $W \cup \mathrm{cons}(\Pi[d]) \models \mathrm{prec}(d)$ holds for every $d \in \Pi$.

Definition 12 *(Local and global applicability).* A default d is *locally applicable* to a sequence of defaults Π if $\mathrm{cons}(\Pi) \cup W \models \mathrm{prec}(d)$ and $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(d)$. Global applicability also requires $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(\Pi \cdot [d])$.

The default logic semantics considered in this article can be defined from the concept of process. In particular, each semantics defines some processes to be maximal sequences of defaults that can be applied. These processes are called "selected", and are defined as follows.

Definition 13 *(Selected process).* Processes are selected if:

Reiter [44]: a process Π is selected if $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(d)$ for each $d \in \Pi$ and no default $d' \notin \Pi$ is locally applicable to Π;
Justified [33]: a process Π is selected if it is a maximal process such that $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(d)$ for each $d \in \Pi$;
Constrained [11,46]: a process Π is selected if it is a maximal process such that $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(\Pi)$;
Rational [39]: a process Π is selected if $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(\Pi)$ and no default $d' \notin \Pi$ is globally applicable to Π.

The conditions for a process to be selected can all be broken in two parts: success (a consistency condition) and closure (non-extendibility of the process). For example, for constrained default logic the condition of success is $\mathrm{cons}(\Pi) \cup W \top \mathrm{just}(\Pi)$ and the condition of closure is that $\Pi \cdot [d]$ is not a successful process for any $d \notin \Pi$.

For a full rationale of these definitions we refer to the original articles where they have been introduced [11,33,39,44,46], compared [10,29], or surveyed [2,42]. Essentially, Reiter and justified default logics require the justification of each default to be consistent with the background theory and the consequences of the defaults in the process. Constrained and rational default logics instead require the union of all justifications to be consistent with the background theory and the consequences of the defaults in the process. This difference gives one "axis" on which these semantics differ. The other axis is whether a partially built process is allowed to "fail" [29], which is the condition where the addition of a default to a process contradicts the justification of a previously applied default. Reiter and rational default logic allow a process to fail, so that no extension would be generated by the process. Justified and constrained default logic disallow failing: a default cannot be added to a process if that would result in a failure.

As an example, consider the default theory $\langle\{d_1, d_2, d_3\}, \emptyset\rangle$ where $d_1 = \frac{:a \wedge b}{a}$, $d_2 = \frac{a:c}{\neg b \wedge c}$, and $d_3 = \frac{a:\neg b \wedge \neg c}{\neg c}$. In all considered semantics, d_2 and d_3 can only be applied after a has been derived, that is, after d_1 has been applied. Since the conclusion of each default negates the justification of the other, d_2 cannot follow d_3 in a selected process and vice versa. Of the remaining processes $[d_1]$, $[d_1, d_2]$ and $[d_1, d_3]$, the four semantics behave as follows: for Reiter, the first is unclosed, the second fails, and the third is selected; for justified, the first is unclosed, the second is unsuccessful, and the third is selected; for constrained, the first process is selected, the other two are not successful; for rational, the first is unclosed, the second fails, and the third is unsuccessful.

Remarkably, all definitions of selected processes only mention the background theory W in conjunction with $\mathrm{cons}(\Pi)$, that is, W only occurs in subformulae of the form $W \cup \mathrm{cons}(\Pi)$. The only conditions for which this is not true is that of Π being a process. The selected processes of a default theory determine the extensions of the theory, from which entailment is in turn defined.

Definition 14 *(Extension).* An extension of a default theory $\langle D, W\rangle$ is a set $Cn(W \cup \mathrm{cons}(\Pi))$ where Π is a selected process of the theory. We denote by $\mathrm{Ext}_D(W)$ the set of all (finite) formulae that are equivalent to an extension of $\langle D, W\rangle$.

We remind the assumption that the considered alphabet is finite. Including formulae that are equivalent to the extensions in this set allows to write $E \in \mathrm{Ext}_D(W)$ to denote the equivalence of E with an extension of $\langle D, W\rangle$. In the skeptical semantics (also called cautious) a default theory entails a formula if all its extensions do.

Definition 15 *(Skeptical entailment).* A default theory $\langle D, W\rangle$ entails a formula W' if $E \models W'$ for every $E \in \mathrm{Ext}_D(W)$.

This condition is equivalent to $\vee\mathrm{Ext}_D(W) \models W'$ (see the Preliminaries section for the definition of the notation $\vee S$ when S is an infinite set of formulae); as a result, the set of all consequences of a default theory is equivalent to $\vee\mathrm{Ext}_D(W)$. Entailment of W' from a theory $\langle D, W\rangle$ is denoted by $W \models_D W'$. This notation emphasizes that each set of defaults D defines a non-monotonic inference relation $\models_D$ [35]. Equivalence of two default theories is defined as either pairwise propositional equivalence of extensions, or equality of consequences. In the following section we provide formal definitions of entailment and equivalence in default logic.

Some semantics of default logic do not assign any extension to some theories. In this paper, we try to derive existence results (counterexamples and proofs of hardness) using theories having extensions.

5.1. Equivalence in default logics

The definition of redundancy is based on that of equivalence, and in particular on the equivalence of a formula with one of its proper subsets. As explained in the previous section, two forms of equivalence are natural in default logic: one based on extensions, the other on consequences. Remarkably, none of these two definitions of equivalence is related to $\models_D$ by full entailment. We therefore define a new form of equivalence, derived from $\models_D$ by full entailment, and two new forms of entailment, derived from the two aforementioned definitions of equivalence by full entailment.

This leads to three definitions of entailment and three of equivalence. Of these, one definition of entailment and two of equivalence are of actual interest, the other definitions being technical tools. The first form of equivalence is derived from skeptical entailment.

Definition 16 *(Entailment and mutual equivalence).* For a given set of defaults D, formula W entails formula W', denoted by $W \models_D W'$, if $\vee\mathrm{Ext}_D(W) \models W'$. These two formulae are mutually equivalent, denoted by $W \equiv_D^m W'$, if $W \models_D W'$ and $W' \models_D W$.

In classical logic, this definition of equivalence is the same as W and W' having the same set of consequences and the same set of models. In default logic, this is not the case. We define the equivalence based on the set of consequences as follows; an associated definition of entailment is defined for convenience.

Definition 17 *(Consequence-entailment and consequence-equivalence).* For a given set of defaults D, formula W consequence-entails formula W', denoted $W \models_D^c W'$, if $\vee\mathrm{Ext}_D(W) \models \vee\mathrm{Ext}_D(W')$. These two formulae are consequence-equivalent, denoted $W \equiv_D^c W'$, if $\vee\mathrm{Ext}_D(W) \equiv \vee\mathrm{Ext}_D(W')$.

A more stringent condition of equivalence of two defaults theories is that of having the same extensions. Again, an associated entailment relation is defined for technical convenience. The rationale of this definition is that the extensions of a default theory are the possible alternative scenarios that are expressed by the default theory, so in way it is the most faithful representation of what a theory expresses. Other authors have considered default theories to be equivalent when they have the same extensions [10,21,24].

Definition 18 *(Faithful entailment and faithful equivalence).* For a given set of defaults D, formula W faithfully entails formula W', denoted $W \models_D^e W'$, if $\mathrm{Ext}_D(W) \subseteq \mathrm{Ext}_D(W')$. These two formulae are faithfully equivalent, denoted $W \equiv_D^e W'$, if $\mathrm{Ext}_D(W) = \mathrm{Ext}_D(W')$.

In a later section this form of equivalence will be shown the same as consequence-equivalence when the credulous consequences are considered and Reiter semantics is used.

It is easy to see that $W \models_D^e W'$ implies $W \models_D^c W'$, which in turn implies $W \models_D W'$. Other results and counterexamples regarding these forms of entailment will be show in the next section.

Each of the above three equivalence relations is related to the corresponding entailment relation by full entailment: two formulae are equivalent if and only if each formula entails the other. Since redundancy is defined in terms of equivalence, we are especially interested in the natural equivalence relations $\equiv_D^c$ and $\equiv_D^e$, that is, equality of consequences and equality of extensions. Mutual equivalence has been defined for technical reasons.

Redundancy in default logic is defined as follows.

Definition 19 *(Redundancy of a clause).* For a given set of defaults D, a clause γ is redundant in a formula W according to equivalence $\equiv_D^x$ if $W \equiv_D^x W \backslash \{\gamma\}$.

Definition 20 *(Redundancy of a formula).* For a given set of defaults D, a formula W is redundant according to equivalence $\equiv_D^x$ if there exists $W' \subset W$ such that $W \equiv_D^x W'$.

In both cases, we are comparing for equivalence a formula and one of its subsets. In Appendix C we study the equivalence of W' and W when $W' \subseteq W$. Equivalence depends on the semantics (Reiter, justified, constrained, and rational); we try to derive results that hold for all these four semantics. The results can be summarized as follows:

Counterexample 9. *There exists a set of defaults D and two formulae W and W' such that $W' \subset W$, $W' \equiv_D^m W$, and $W' \not\equiv_D^c W$ in Reiter, justified, constrained and rational default logic.*

Counterexample 10. *There exists a set of defaults D and two formulae W and W' such that $W' \subset W$, $W' \equiv_D^c W$, and $W' \not\equiv_D^e W$ in Reiter and justified default logic.*

Counterexample 11. *There exists a set of defaults D and two formulae W and W' such that $W' \subset W$, $W' \equiv_D^c W$, and $W' \not\equiv_D^e W$ in constrained and rational default logic.*

Trivial $W' \models_D^e W \Rightarrow W' \models_D^c W$.

Lemma 7. *For any set of defaults D and two formulae W and W', if $W' \models_D^c W$ then $W' \models_D W$, in Reiter, justified, constrained and rational default logic.*

Lemma 8. *For any set of defaults D and two formulae W and W', if $W' \subseteq W$ and $W' \models_D W$ then $W' \models_D^e W$ for Reiter, justified, constrained and rational default logic.*

Corollary 4. *For any set of defaults D and two formulae W and W', if $W' \subseteq W$, then in Reiter, justified, constrained, and rational default logic the following holds:*

$$W' \models_D W \Leftrightarrow W' \models_D^c W \Leftrightarrow W' \models_D^e W$$

Theorem 16. *For any set of normal defaults D and two formulae W and W', if $W' \subseteq W$ then $W' \equiv_D^c W$ implies $W' \equiv_D^e W$ in constrained default logic.*

Theorem 17. *If D is a set of normal defaults and W and W' are two formulae such that $W' \subseteq W$, then $W' \equiv_D^c W$ holds if and only if $W' \equiv_D^e W$ holds for Reiter, justified, constrained, and rational default logic.*

Lemma 9. *If D is a set of categorical defaults and W and W' are two formulae such that $W' \subseteq W$ and $W' \models_D W$, then $W \models_D^e W'$ in constrained default logic.*

Corollary 5. *If D is a set of normal and categorical defaults and W and W' are two formulae such that $W' \subseteq W$, the conditions $W' \equiv_D^m W$, $W' \equiv_D^c W$, and $W' \equiv_D^e W$ are equivalent in Reiter, justified, constrained, and rational default logic.*

5.2. Redundancy of clauses vs. theories

In this section, we show that default logic does not have the unitary redundancy property in general, but it has it if all defaults are categorical and normal.

Counterexample 3. *There exists a set of normal defaults D and two formulae W and W' such that $W' \subset W$, W is equivalent to W', and $W \backslash \{\gamma\}$ is not equivalent to W for any $\gamma \in W$, for mutual, faithful, and consequence equivalence in Reiter, justified, constrained, and rational default logic.*

Proof. The defaults D and background theory W are as follows.

$$W = \{a, b\}$$

$$W' = \emptyset$$

$$D = \{d_1, d_2, d_3\}$$

where

$$d_1 = \frac{a : \neg b}{\neg b}$$

$$d_2 = \frac{b : \neg a}{\neg a}$$

$$d_3 = \frac{: a \wedge b}{a \wedge b}$$

We show that $\langle D, W\rangle$ and $\langle D, \emptyset\rangle$ have the same extensions, thus proving equivalence of W and W' according to all three definitions. We also show that neither $\vee\mathrm{Ext}_D(\{a\})$ nor $\vee\mathrm{Ext}_D(\{b\})$ entail W, thus proving their non-equivalence to W for all three definitions.

The theory $\langle D, W\rangle$ has the single extension $Cn(\{a, b\})$. Indeed, d_1 and d_2 are not applicable because their justifications are inconsistent with W. The third default is applicable, but its consequence is $a \wedge b$, which is already in the theory.

The theory $\langle D, \emptyset\rangle$ has the same single extension: d_3 is the only applicable default, leading to the addition of $a \wedge b$, which makes both d_1 and d_2 inapplicable. Therefore, $Cn(\{a, b\})$ is the only extension of this theory. Since $\langle D, W\rangle$ and $\langle D, \emptyset\rangle$ have the same extensions, they are equivalent according to all three definitions of equivalence.

The theory $\langle D, \{a\}\rangle$ still has the extension $\{a, b\}$, which results from the application of d_3, which then blocks the application of d_1 and d_2. However, it also has a new extension: since d_1 is applicable, it generates $\neg b$, which blocks the application of d_3. This produces the extension $Cn(\{a, \neg b\})$. Therefore, $\vee\mathrm{Ext}_D(\{a\})$ is equivalent to a, and therefore does not entail W. As a result, $\{a\}$ is not equivalent to W for any of the three definitions of equivalence.

In the same way, $\langle D, \{b\}\rangle$ has the two extensions $Cn(\{a, b\})$ and $Cn(\{\neg a, b\})$, and $\vee\mathrm{Ext}_D(\{b\})$ is equivalent to b, which does not entail W. □

Since unitary redundancy does not hold, monotonic redundancy does not hold either. This means that the preconditions of Theorems 1, 2 and 3, are not satisfied. The precondition of the second and third theorem are not satisfied because the considered semantics of default logic are neither monotonic nor cumulative. The following theorems show which preconditions of the other theorem are not satisfied.

Counterexample 4. *Mutual equivalence is not transitive for Reiter, justified, constrained, and rational default logic, even for normal defaults.*

Proof. We show a set of defaults D such that $\{a\} \equiv_D^m \emptyset$, $\emptyset \equiv_D^m \{a, b\}$ but $\{a\} \not\equiv_D^m \{a, b\}$. This set of defaults is as follows.

$$D = \left\{ \frac{a : \neg b}{\neg b}, \frac{: a \wedge b}{a \wedge b} \right\}$$

The extensions are as follows: $\emptyset$ and $\{a, b\}$ have the single extension $Cn(\{a, b\})$, since the second default is the only applicable one, and its consequence prevents the application of the first one; $\{a\}$ has two extensions $Cn(\{a, \neg b\})$ and $Cn(\{a, b\})$, since both defaults are applicable but the consequence of each one prevent the application of the other.

Since $Cn(\{a, b\})$ entails $\{a\}$ and $\{a, b\}$, while $\emptyset$ is entailed by all considered extensions, we have that $\{a\} \equiv_D^m \emptyset$ and $\emptyset \equiv_D^m \{a, b\}$. On the other hand, $\vee \mathrm{Ext}_D(\{a\}) \equiv \{a\}$, which does not entail $\{a, b\}$. As a result, $\{a\} \not\equiv_D^m \{a, b\}$. □

Remarkably, consequence-equivalence and faithful equivalence are instead transitive by definition. The property that does not hold is right weakening of their corresponding entailment relations.

Counterexample 5. *Consequence-entailment and faithful entailment do not have the right weakening property in Reiter, justified, constrained, and rational default logic, and even for normal defaults.*

Proof. This is shown by $D = \{\frac{:\neg a}{\neg a}\}$ and the two formulae $W = \{a\}$ and $W' = \emptyset$. Clearly, $W \models_D^e W$ and $W \models_D^c W$. On the other hand, the only extension of W' is $Cn(\{\neg a\})$. Since the set of the extensions of W is not contained in the set of extensions of W', the first formula does not faithfully entail the second, that is, $W \not\models_D^e W'$. For the same reason, consequence-entailment does not hold either. □

Interestingly, this counterexample involves a theory that is both categorical and normal. Theorem 1 cannot be used because consequence-entailment and faithful entailment do not meet its precondition. However, monotonic redundancy can still be proved for categorical and normal defaults.

Theorem 10. *Mutual equivalence, faithful equivalence, and consequence equivalence have the monotonic redundancy property if all defaults are categorical and normal in Reiter, justified, constrained, and rational default logic.*

Proof. If all defaults are categorical and normal, the three forms of equivalence coincide. Since faithful equivalence is transitive, mutual equivalence is transitive as well in this case. Since $\models_D$ is reflexive, has right weakening, and is related to mutual equivalence by full-entailment, Theorem 1 applies: mutual equivalence has the monotonic redundancy property in this case. As a result, all three forms of equivalence have this property if all defaults are categorical and normal. □

Incidentally, Corollary 5 only states that the three forms of equivalence coincide for categorical and normal defaults, not that the same applies to the corresponding forms of entailment. This explains why faithful entailment and consequence-entailment do not have the right weakening property, which entailment has, even in the case of normal and categorical defaults.

5.3. Making clauses irredundant

Modifying a formula in order to make some of its parts irredundant proved useful for classical and circumscriptive logics. We show a similar result for default logic. Such results are generally useful to build a theory having some properties regarding redundancy. For example, one can reduce the problem of redundancy of a clause to the problem of redundancy of a theory, as checking whether γ is redundant in Π is the same as checking the redundancy of γ in the modified version of Π in which all clauses but γ are made irredundant.

Definition 21. The M-irredundant version of a default theory $\langle D, W\rangle$, where $M \subseteq W = \{\gamma_1, \ldots, \gamma_m\}$, is the following default theory, where $\{d, c_1, \ldots, c_m\}$ are new variables.

$$I(\langle D, W\rangle, M) = \langle D', W'\rangle$$

where:

$$W' = \{c_i \vee \gamma_i \mid 1 \leqslant i \leqslant m\}$$
$$D' = D_1 \cup D_2 \cup D_3$$
$$D_1 = \left\{ \frac{: \neg d}{\neg d}, \frac{\neg d : \neg c_1 \wedge \cdots \wedge \neg c_m}{\neg c_1 \wedge \cdots \wedge \neg c_m} \right\}$$
$$D_2 = \left\{ \frac{\neg c_1 \wedge \cdots \wedge \neg c_m \wedge \alpha : \beta}{\gamma} \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D \right\}$$
$$D_3 = \left\{ \frac{c_i \vee \gamma_i : \neg c_i \wedge d \wedge \{c_j \mid 1 \leqslant j \leqslant m,\ j \neq i\}}{\neg c_i \wedge d \wedge \{c_j \mid 1 \leqslant j \leqslant m,\ j \neq i\}} \,\middle|\, 1 \leqslant i \leqslant m, \gamma_j \in M \right\}$$

The clauses of M are made irredundant by this transformation, while the redundancy of the other clauses does not change.

Theorem 11. *For any default theory $\langle D, W \rangle$ and formulae M and W'', if $M \subseteq W$, $\langle D', W' \rangle = I(\langle D, W \rangle, M)$, and $W'' \subset W$, then $W' \equiv^e_{D'} W''$ holds if and only if $\{\gamma_i \mid c_i \vee \gamma_i \in W''\} \equiv^e_D W$ and W'' contains all clauses $c_i \vee \gamma_i$ such that $\gamma_i \in M$. This holds for Reiter, justified, constrained and rational default logic, and also for consequence-equivalence.*

Proof. The first default of D_1 can always be applied to W', leading to $\neg d$, which makes all defaults of D_3 inapplicable. If W is inconsistent, no other default can be then applied. Otherwise, the only applicable default is the second one of D_1, which generates $\neg c_1 \wedge \cdots \wedge \neg c_m$. This formula makes W' equivalent to W and the defaults of D_2 equivalent to those of D. As a result, $I(\langle D, W \rangle, M)$ has an extension $Cn(E \cup \{\neg d, \neg c_1, \ldots, \neg c_m\})$ for any extension E of $\langle D, W \rangle$, if W is consistent, and the extension $Cn(\{d\})$ otherwise. A subset of W' has such an extension if and only if the corresponding subset of W has the same extension modulo the new variables. This means that if a clause is irredundant in $\langle D, W \rangle$, the corresponding clause of $\langle D', W' \rangle$ is irredundant as well.

We now prove that all clauses $c_i \vee \gamma_i$ such that $\gamma_i \in M$ are irredundant as well. This is proved by showing that for each clause of M the corresponding default of D_3 can only be applied if that clause is in the background theory, and that its effect is to produce an extension that cannot generated otherwise.

Every default of D_3 is applicable to W: the ith default is applicable because its precondition $c_i \vee \gamma_i$ is in the background theory, and its justification is consistent with the background theory, since $W' \cup \{d, \neg c_i\} \cup \{c_j \mid \ldots\}$ is equivalent to $\{\gamma_i, d, \neg c_i\} \cup \{c_j \mid \ldots\}$. The application of such a default makes all other defaults of D' inapplicable.

This means that the ith default of D_3 is not applicable in a subset of W' that does not contain $c_i \vee \gamma_i$: the other clauses of W do entail this one because each clause contains a different c_i, and the application of any other default blocks this one.

Since the extension generated by applying this default is inconsistent with all other extensions of the theory, any subset of W' not containing $c_i \vee \gamma_i$ necessarily has a different set of extensions and consequences than W'.

Finally, if a clause γ_i is redundant in $\langle D, W \rangle$ but does not occur in M, the corresponding clause $c_i \vee \gamma_i$ does not affect the extensions obtained by applying the defaults in D_3. Its removal could therefore only affect the extensions obtained by applying the defaults of $D_1 \cup D_2$; the application of these defaults has however essentially the same effect as the application of the defaults of D in the original theory, which proves that $c_i \vee \gamma_i$ is redundant. □

5.4. Complexity of redundancy in default logics

In this section, we summarize the results about the complexity of checking the redundancy of a clause in the background theory of a default theory and that of checking the redundancy of the background theory.

The following results are about the redundancy for faithful equivalence.

Theorem 18. *Checking whether $W' \equiv^e_D W$ for a set of defaults D and two formulae W and W' such that $W' \subseteq W$ is in Π^p_2 for Reiter and justified default logic and in Π^p_3 for constrained and rational default logic.*

Theorem 19. *Checking whether $W' \equiv_D^e W$, where D is a set of defaults and W and W' are two formulae, is Π_2^p-hard even if $W = W' \cup \{a\}$ where a is a variable, W is consistent, and all defaults are categorical and normal, in Reiter, justified, constrained, and rational default logic.*

Theorem 20. *Checking whether $W \equiv_D^e W'$, where D is a set of defaults and W and W' two formulae, is Π_3^p-hard for constrained and rational default logic even if $W = W' \cup \{a\}$, where a is a variable.*

As a corollary, we can conclude that the problems of redundancy of a clause and that of redundancy of the background theory under faithful equivalence are Π_2^p-complete for Reiter and justified default logic and Π_3^p-complete for constrained and rational default logic.

We now consider the problem of redundancy of clauses when consequence-equivalence is used. The difference between the two kinds of equivalence is that two sets of extensions may be different but yet their disjunctions are the same. The necessity of calculating the disjunction of all extensions intuitively explains why checking redundancy for consequence-equivalence can be harder than for faithful equivalence.

Theorem 21. *Checking whether $W' \equiv_D^c W$, where D is a set of defaults and W and W' are two formulae, is in Π_3^p if $W' \subseteq W$ for Reiter, justified, constrained, and rational default logic.*

Theorem 22. *The problem of checking whether $W' \equiv_D^c W$, where D is a set of default and W and W' two formulae, is Π_3^p-hard even if $W = W' \cup \{a\}$ where a is a variable for Reiter, justified, rational, and constrained default logic.*

Corollary 6. *The problem of checking whether $W' \equiv_D^c W$, where D is a set of default and W and W' two formulae, is Π_3^p-complete if $W' \subseteq W$; hardness holds even if $W = W' \cup \{a\}$ where a is a variable. This holds for Reiter, justified, constrained, and rational default logic.*

The next problem to analyze is whether a formula is redundant, still for a fixed set of defaults. The results proved in the appendix are as follows.

Theorem 23. *The problem of formula redundancy for faithful and consequence-equivalence is in Σ_3^p and Σ_4^p, respectively, for Reiter, justified, constrained, and rational default logic.*

Theorem 24. *The problem of formula redundancy based on faithful equivalence is Σ_3^p-hard, and remains hard for consistent theories, for Reiter, justified, constrained, and rational default logic.*

In order to prove the Σ_4^p-hardness of the problem of formula redundancy under consequence-equivalence, we should provide a reduction from ∃∀∃∀QBF validity into this problem. A simpler proof can however be given, based on the following consideration: checking clause redundancy has been proved Π_2^p-hard or Π_3^p-hard using reductions from QBFs that results in default theories having $W = \{a\}$ as the background theory. As a result, these reductions also prove that formula redundancy is Π_2^p-hard or Π_3^p-hard. In other words, we can reduce the validity of a ∀∃QBF or a ∀∃∀QBF into the problems of formula redundancy. What we show is that, if such reductions satisfy some assumptions, we can obtain new reductions from QBFs having an additional existential quantifier in the front. The assumptions are that the default theory resulting from the reduction is such that:

1. the background theory that results from the reduction is classically irredundant;
2. the matrix of the QBF occurs only in the justification of a single default and does not affect the rest of the default theory.

The reductions used for proving the hardness of clause redundancy satisfy both assumptions. In particular, $\forall X \exists Y \forall Z.F$ is valid if and only if the background theory of the following theory is consequence-redundant, where $W, D, \alpha, \beta, \gamma$, do not depend on F but only on the quantifiers of the QBF and W is classically irredundant.

$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F}{\gamma} \right\}, W \right\rangle$$

The fact that the matrix of the QBF is copied "verbatim" in the default theory is exploited as follows: if ω_w is a truth evaluation over the variable w, then $\forall X \exists Y \forall Z . F|_{\omega_w}$ is valid if and only if the background theory of $\langle D \cup \{\frac{\alpha:\beta \wedge F \wedge \omega_w}{\gamma}\},\ W\rangle$ is redundant. This default theory can be modified in such a way that the subsets of the background theory are in correspondence with the truth evaluations over ω_w. This way, the resulting theory is redundant if and only if $\exists w \forall X \exists Y \forall Z\ .\ F$ is valid. The resulting default theory still satisfies the two assumptions above on the background theory and on the use of the matrix of the QBF; therefore, this procedure can be iterated to obtain a reduction from ∃∀∃∀QBF validity into the problem of formula redundancy under consequence-equivalence. A similar technique can be used for faithful equivalence.

The details of this technique are reported in the appendix. The results obtained this way are as follows.

Theorem 25. *Formula redundancy is Σ^p_3-hard for faithful equivalence and Σ^p_4-hard for consequence-equivalence in Reiter, justified, constrained, and rational default logic.*

5.5. Credulous default reasoning

In the credulous approach to default logic, if a formula is in at least one extension of the theory, it is entailed by the theory. Contrarily to skeptical reasoning, presence in all extensions is not required; presence in one suffices. Credulous equivalence can be defined in the obvious way: two theories (or two background theories, in the assumption that a fixed set of defaults is considered) are equivalent if they have the same credulous consequences.

If two theories are faithfully equivalent, they also have the same consequences, regardless of whether the skeptical or credulous approach is used. The converse has already been proved untrue for the skeptical semantics. For the credulous semantics, things are slightly more complicated: credulous equivalence implies faithful equivalence for Reiter default logic but not for the other three considered semantics.

Theorem 12. *If two default theories have the same credulous consequences under Reiter default logic, they also have the same extensions.*

Proof. Let us assume to the contrary that E is an extension of T but not of T', where T and T' have the same credulous consequences. Let F be a finite formula equivalent to E. By definition, F is entailed by T under the credulous semantics. Therefore, F is also entailed by T'. This means that T' has an extension E' such that $E' \models F$, which implies $E \subseteq E'$.

Let F' be a finite formula that is equivalent to E'. We have that F' is entailed by T', and is therefore also entailed by T. As a result, T has an extension E'' such that $E'' \models F'$. This implies $E'' \models F$, and therefore $E'' \models E$, which means $E \subseteq E''$.

Under Reiter semantics, two extensions of the same theory cannot be one strictly contained into the other [44]. This implies $E = E''$, and therefore $E = E'$. Since E' is an extension of T', this contradicts the assumption that E is not an extension of T'. □

This theorem shows that, in Reiter default logic, the results about the redundancy of default theories under faithful equivalence also hold for credulous equivalence.

The proof of this theorem is based on the impossibility for a theory to have an extension strictly contained in another. This is also true for QDL when the background theory has empty support [19] and for ZDL [27], where the above theorem therefore holds.

This condition is however not true for all semantics. We can indeed show a counterexample where two theories have different extensions but the same credulous consequences. This counterexample involves a theory having an extension contained into another.

Counterexample 6. *There exists D, W, $W' \subseteq W$ such that $\langle D, W\rangle$ and $\langle D, W'\rangle$ have the same credulous consequences but different sets of extensions under justified, constrained and rational default logic.*

Proof. The statement holds for $W = \{a\}$, $W' = \emptyset$ and D defined as follows.

$$D = \left\{ \frac{: a \wedge b}{a \wedge b}, \frac{: \neg a \wedge b}{b} \right\}$$

In all three considered semantics, the first default can be applied to both W and W', leading to $a \wedge b$ which blocks the application of the second default. Both theories therefore have the extension $Cn(\{a \wedge b\})$.

The second default cannot be applied to W because W contradicts its justification. On the other hand, it can be applied to W', generating b. The first default cannot be now applied, for different reasons: in constrained and rational default logic, the justifications of the two defaults contradict each other; in justified default logic, the first default could be applied, but that would lead to a failure because its consequence $a \wedge b$ contradicts the justification $\neg a \wedge b$ of the previously applied second default. As a result, W' has the additional extension $Cn(\{b\})$.

Since $\langle D, W\rangle$ and $\langle D, W'\rangle$ have different sets of extensions, they are not faithfully equivalent. On the other hand, they have exactly the same credulous consequences. Indeed, the only additional extensions of the second theory is $Cn(\{b\})$, whose consequences are all contained in the consequences of the common extension $Cn(\{a \wedge b\})$. □

5.6. Redundancy of defaults

For the sake of simplicity, we analyze the redundancy of defaults only for the case of faithful equivalence. For a set of clauses W and defaults D, we define $\mathrm{Ext}_W(D)$ to be the set of extensions of $\langle D, W\rangle$, that is, $\mathrm{Ext}_W(D) = \mathrm{Ext}_D(W)$. The definitions of faithful entailment and equivalence are defined as follows.

Definition 22 *(Faithful equivalence of sets of defaults).* Given a fixed background theory W, a set of defaults D faithfully entails a set of defaults D', written $D \models^e_W D'$, if $\mathrm{Ext}_W(D) \subseteq \mathrm{Ext}_W(D')$. These two sets are faithfully equivalent, denoted by $D \equiv^e_W D'$, if $D \models^e_W D'$ and $D' \models^e_W D$.

For a fixed background theory, we can define redundancy of a default in a set of defaults as the equivalence of the set with the set where the default has been removed.

Definition 23 *(Redundancy of a default).* A default d is redundant in D with respect to a background theory W if and only if $D \backslash \{d\}$ is faithfully equivalent to D.

Redundancy of sets of defaults is defined as follows.

Definition 24 *(Redundancy of a theory).* A set of defaults D is redundant with respect to a background theory W if there exists $D' \subset D$ such that D' is faithfully equivalent to D.

5.6.1. Making defaults irredundant

The following lemma is the version of Theorem 11 in the case of default redundancy rather than clause redundancy. It proves that some defaults can be made irredundant while not changing the redundancy status of the others. Such results are generally useful to build default theories having some specific properties w.r.t. redundancy. In particular, the following lemma will be later used in Theorem 14 to prove that the problems of redundancy of clauses can be reduced to the corresponding problems of redundancy of defaults.

Lemma 5. *For every default theory $\langle D, W\rangle$ and every set of defaults $D_I \subseteq D$, let p and q two new variables and $\{v_i\}$ be a set of new variables in bijective correspondence with the defaults of D_I, and let D_1, D_2, D_3 be defined as follows:*

$$D_1 = \{d+, d-\}$$

where:

$$d+ = \frac{: p \wedge q}{p \wedge q}$$

$$d- = \frac{:\neg p \wedge q}{\neg p \wedge q}$$

$$D_2 = \left\{ \frac{q \wedge (\neg p \vee \alpha) : \neg p \vee \beta}{(p \vee v_i) \wedge (\neg p \vee \gamma)} \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D_I,\ v_i \textit{ is the variable corresponding to } \frac{\alpha : \beta}{\gamma} \right\}$$

$$D_3 = \left\{ \frac{q \wedge p \wedge \alpha : \beta}{\gamma} \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D \backslash D_I \right\}$$

if $\langle D, W\rangle$ has extensions and W is consistent, in Reiter, justified, constrained, and rational default logics it holds that:

1. *the processes of $\langle D_1 \cup D_2 \cup D_3, W\rangle$ are (modulo the transformation of the defaults) the same as of $\langle D, W\rangle$ with $d+$ added to the front and a number of processes composed of $d-$ and a sequence containing all defaults of D_2*;
2. *the extensions of $\langle D_1 \cup D_2 \cup D_3, W\rangle$ are the same as of $\langle D, W\rangle$ with $\{p, q\}$ added plus the single extension $\{\neg p, q\} \cup \{v_i\}$*;
3. *a subset of $D_1 \cup D_2 \cup D_3$ is faithfully equivalent to it if and only if it contains $D_1 \cup D_2$ and the set of original defaults corresponding to those of $D_2 \cup D_3$ is faithfully equivalent to D.*

5.6.2. Redundancy of defaults vs. sets of defaults

While a formula is classically redundant if and only if it contains a redundant clause, the same does not happen for default redundancy. The following counterexamples indeed show that Reiter, constrained, and rational default logic do not have the unitary redundancy property w.r.t. redundancy of defaults.

Counterexample 7. *There exists a set of defaults D such that, according to Reiter and rational default logic*:

1. *for every $d \in D$, the theory $\langle D\backslash\{d\}, \emptyset\rangle$ has extensions and $D\backslash\{d\} \not\equiv^e_\emptyset D$*;
2. *there exists $D' \subset D$ such that $D' \equiv^e_\emptyset D$.*

Counterexample 8. *There exists a set of defaults D such that, according to constrained default logic*:

1. *for every $d \in D$, it holds $D\backslash\{d\} \not\equiv^e_\emptyset D$*;
2. *there exists $D' \subset D$ such that $D' \equiv^e_\emptyset D$.*

Justified default logic has the unitary redundancy property w.r.t. default redundancy. This is a combination of two factors: first, justified default logic is failsafe [29] (every successful process can be made selected by adding some defaults); second, every extension is generated by an unique set of defaults.

Theorem 13. *Faithful equivalence of sets of defaults for justified default logic has the monotonic redundancy property.*

We therefore have as a corollary that justified default logic has the unitary redundancy property when redundancy of defaults is considered.

Corollary 2. *Faithful equivalence of sets of defaults under justified default logic has the unitary redundancy property.*

Remarkably, monotonic redundancy could not be proved by using Theorems 1, 2, or 3. Indeed, faithful entailment and equivalence for justified default logic do not satisfy the preconditions of any of these theorems. Right weakening does not hold as shown by $W = \emptyset$, $D = \{\frac{:a}{a}, \frac{:\neg a}{\neg a}\}$, $D' = D$, and $D'' = \emptyset$: while $\mathrm{Ext}(D) \subseteq \mathrm{Ext}(D')$ obviously holds (i.e., D entails D') and $D'' \subseteq D'$, the set D has the extension $Cn(\{a\})$ which D'' does not have; therefore, $\mathrm{Ext}(D) \not\subseteq \mathrm{Ext}(D'')$, that is, D does not entail D''. Monotonicity does not hold either: it can be the case that $\mathrm{Ext}(D) \subseteq \mathrm{Ext}(D')$ and $D \subseteq D''$, but $\mathrm{Ext}(D'') \not\subseteq \mathrm{Ext}(D')$, as for example for $W = \emptyset$, $D = D' = \{\frac{:a}{a}\}$ and $D'' = D \cup \{\frac{:b}{b}\}$.

5.6.3. Complexity of redundancy of defaults

For Reiter default logic, an upper bound on complexity can be given by showing a reduction from the complexity of clause or formula redundancy to the corresponding problems for defaults. This is possible thanks to the following lemma.

Lemma 6. *If $W \cup \{\gamma\}$ is a consistent formula and D a set of defaults, $\langle D, W \cup \{\gamma\}\rangle$ and $\langle D \cup \{\frac{:\top}{\gamma}\}, W\rangle$ have the same Reiter extensions.*

Proof. We prove that every selected process of the first theory can be turned into a selected process of the second theory by the addition of the default $\frac{:\top}{\gamma}$, and vice versa. Let Π be a selected process of $\langle D, W \cup \{\gamma\}\rangle$. Since $\frac{:\top}{\gamma}$ has no precondition, it is always applicable. As a result, $[\frac{:\top}{\gamma}] \cdot \Pi$ is a process of $\langle D \cup \{\frac{:\top}{\gamma}\}, W\rangle$. In order to prove that this is a selected process, we observe that the background theory and the consequences of the default of a process always occur together in the definition of a selected process. As a result, it does not matter if γ is in the background theory or in consequence of a default in a process.

Let us now assume that Π is a selected process of the second theory. Since $\frac{:\top}{\gamma}$ has no precondition, it is in Π, as otherwise this process would not be closed. Let Π' be the process obtained by removing $\frac{:\top}{\gamma}$ from Π. This is a selected process of $\langle D, W \cup \{\gamma\}\rangle$, because the lack of the clause γ from $\mathrm{cons}(\Pi')$ is compensated by the presence of γ in W. $\square$

If a background theory is consistent, this lemma can be applied to all its clauses, leading to the following characterization of the complexity of redundancy for defaults.

Theorem 14. *The problem of redundancy of a default and of a default theory for faithful equivalence in Reiter default logic are Π_2^p-hard and Σ_3^p-hard, respectively.*

Proof. Hardness of clause and formula redundancy remains the same even if the formula is consistent by Theorem 19 and Theorem 24. As a result, Lemma 6 can be applied: a clause γ is redundant in $\langle D, W\rangle$ if and only if d_γ is redundant in $\langle D \cup \{d_\gamma\}, W\backslash\{\gamma\}\rangle$. Indeed, $\langle D, W\rangle$ has the same extensions of $\langle D \cup \{d_\gamma\}, W\backslash\{\gamma\}\rangle$, and removing γ from the first theory or removing d_γ from the second theory lead both to $\langle D, W\backslash\{\gamma\}\rangle$.

The problem of formula redundancy can be reduced to default redundancy by first applying Lemma 6 to all clauses of W, and then making all original defaults irredundant using the transformation of Lemma 5. $\square$

In order to show membership of the problems of redundancy, we prove that faithful equivalence of sets of defaults is in Π_2^p.

Theorem 15. *Checking faithful equivalence of two sets of defaults is in Π_2^p for Reiter default logic.*

Proof. The contrary of the statement amounts to checking whether any of the two theories have an extension that the other one does not have. The number of possible extensions, however, is limited by the fact that any extension is generated by the set of consequences of some defaults.

Checking whether $\langle D, W\rangle$ has an extension that $\langle D', W'\rangle$ has not can be done as follows: guess a subset $D'' \subseteq D$, and let $E = \mathrm{cons}(D'')$; check whether E is an extension of $\langle D, W\rangle$ but is not an extension of $\langle D', W'\rangle$.

Checking whether a formula E is an extension of a default theory can be done with a logarithmic number of satisfiability tests [28,45]. As a result, the problem can also be expressed as a QBF formula $\exists\forall$QBF. In order to check whether there exists D'' such that $E = \mathrm{cons}(D'')$ is in this condition, we only have to add an existential quantifier to the front of this formula. The problem is therefore in Π_2^p. $\square$

The problem of checking the redundancy of a set of defaults is obviously in Σ_3^p, as it can be solved by guessing a subsets of defaults and then checking equivalence. As a result, we have a precise complexity characterization of the problems of redundancy of Reiter default logics according to faithful equivalence.

Corollary 3. *The problem of checking the redundancy of a default and the redundancy of a set of defaults are Π_2^p-complete and Σ_3^p-complete, respectively, for Reiter default logic and faithful equivalence.*

6. Conclusions

In this article, some properties of redundancy in some non-monotonic logics has been proved. These properties fall in three categories: some logics have the monotonic or unitary redundancy property while others do not; some parts of a theory can be made irredundant; and results characterizing the complexity of checking the redundancy of an element in a set and the redundancy of a set.

The results can be summarized as follows.

Monotonic/unitary redundancy: hold for circumscription (with all variables minimized), for the redundancy of the background theory for categorical normal default logic, and for the redundancy of defaults in justified default logic; do not hold for the redundancy of the background theory in default logic in general and for the redundancy of defaults for Reiter, constrained, and rational default logic;

Making parts irredundant: for every of the considered formalisms, we were able to provide a transformation that makes an arbitrary part of a set irredundant, while not changing the redundancy status of the other elements of the set;

Complexity results: complexity results regarding the second and third level of the polynomial hierarchy have been provided; the following list gives the results for problem of checking the redundancy of an element.

- CIRC-redundancy (all variables minimized) of a clause is Π_2^p-complete;
- redundancy of a clause of the background theory under faithful equivalence is Π_2^p-complete for Reiter and justified default logic and Π_3^p-complete for constrained and rational default logic;
- redundancy of a clause of the background theory under (skeptical) consequence-equivalence is Π_3^p-complete for Reiter, justified, constrained and rational default logic;
- redundancy of a default under faithful equivalence for Reiter default logic is Π_2^p-complete.

The complexity of checking the redundancy of a set is in the following list.

- CIRC-redundancy (all variables minimized) of a formula is Π_2^p-complete;
- redundancy of a background theory is Σ_3^p-complete for faithful equivalence for Reiter, justified, constrained, and rational default logic;
- redundancy of a background theory is Σ_4^p-complete for (skeptical) consequence-equivalence for Reiter, justified, constrained, and rational default logic;
- redundancy of a set of defaults under faithful equivalence for Reiter default logic is Σ_3^p-complete.

In a way, circumscription with all variables minimized can be seen as a very basic form of non-monotonic reasoning. However, it somehow expresses the core principle of circumscription, as general circumscription can be translated into it [8,9]. Nevertheless, the results about redundancy in this restriction do not necessarily extend to the general case, as the elimination of varying and fixed variable modify the formula to be minimized. The extension of the results presented in this article to the general case is therefore left as an open problem.

Two results are proved regarding credulous default reasoning: equality of credulous consequences coincide with faithful equivalence for Reiter default logic but not for the three other considered semantics. This means that all results regarding faithful equivalence in Reiter default logic extend to credulous equivalence. The problem of whether they hold for the other three semantics is left as an open problem.

While most of the results are about the redundancy of a set of clauses, some results have also been given about the redundancy of a set of defaults. This scenario is interesting because redundancy is defined on something which is not a propositional formula, so the usual propositional entailment relation and connectives cannot be used. For this reason, in the section about general results we have considered the redundancy of an arbitrary set of elements, without restricting these elements to be clauses or propositional formulae.

The complexity results show that the problems of redundancy of a clause in a formula is the same as for the corresponding problem of entailment for all considered cases. While membership to the same class is obvious, hardness is not and required a separate proof. Regarding the relative complexity of clause redundancy and formula redundancy, one may observe that the difference is affected by the unitary redundancy property. If the unitary redundancy property holds, redundancy of a formula is equivalent to the existence of a redundant clause in it; this means that the check for redundancy of the formula can be done by performing a linear number of clause redundancy checks, and these can be parallelized, thus leading the same complexity. On the other hand, if unitary redundancy does not hold, redundancy of a formula requires checking the existence of an equivalent subset of the formula. This corresponds to an additional existential quantification when these tests are expressed as QBFs, which may imply a jump of one level up in the polynomial hierarchy.

An open question is about the extension of the results presented in this article to strong and uniform equivalence. As it has been noted in the study of default logic, redundancy is defined in terms of equivalence, and its property depends on the definition of equivalence used. Two forms of equivalence that have been recently defined are those of strong and uniform equivalence, which can be defined as "update-resistant equivalence" [12,32,50,51]. A clause would therefore be strongly CIRC-redundant not only if they can be removed from a formula without changing the minimal models of the formula, but also if the same happens in the formulae obtained by adding other clauses.

Acknowledgements

The author thanks Jiang Zhengjun for his comments about a previous version of this paper. Thanks also goes to the anonymous reviewers, and in particular for their suggestions of relating unitary redundancy with cumulativity, which lead the author to the study of redundancy in arbitrary logics.

Appendix A. Two counterexamples

The following two counterexamples are about redundancy in arbitrary logics. In this section we use Π, Π', etc. to denote sets of formulae (rather than default processes) to the sake of coherence with the section on arbitrary logics. The first counterexample shows that unitary redundancy does not imply monotonic redundancy (the converse has been proved to hold).

Counterexample 1 *(proof). There exists a logic where $\models_L$ is reflexive, $\equiv_L$ is symmetric and transitive, they are related by full-entailment, and unitary redundance holds, but monotonic redundancy does not.*

Proof. We show that the statement holds for $\models^e_D$ and $\equiv^e_D$ defined by the following set of defaults:

$$D = \left\{ \frac{: x \wedge y}{x \wedge y}, \frac{y : \neg x \wedge y}{\neg x \wedge y} \right\}$$

Monotonic redundancy does not hold: let $\Pi = \{x, y\}$, $\Pi' = \emptyset$, and $\Pi'' = \{y\}$. Both Π and Π' have $Cn(\{x, y\})$ as their only extension, as the first default is the only one that can be applied, and the second default cannot then be applied. In the theory Π'', either the first or the second default can be applied, leading to two extensions: $Cn(\{x, y\})$ and $Cn(\{\neg x, y\})$.

Reflexivity, symmetry, transitivity and full-entailment are obvious from definition. Proving that this logic has the unitary redundancy property is more involved. Most of the theories have $Cn(\{x, y\})$ as their only extension. The only exceptions are the theories where the second default can be applied and the theories where the first default cannot be applied. The theories in the first category entail y but not x; the theories in the second category entail $\neg x \vee \neg y$.

1. $\{\neg x, y\}$ has extension $Cn(\{\neg x, y\})$;
   $\{y\}$ has extensions $Cn(\{x, y\})$ and $Cn(\{\neg x, y\})$;
2. $\{\bot\}$, $\{\neg x \vee \neg y\}$, $\{\neg x\}$, $\{\neg y\}$, $\{\neg x, \neg y\}$, $\{x, \neg y\}$, $\{x \vee y, \neg x \vee \neg y\}$, all have their deductive closure as their only extension.

Each of these theories has a set of extensions that is different from that of the another, and is different from the set containing only $Cn(\{x, y\})$. As a result, if two sets Π and Π' are not faithfully equivalent, then at least one of them is one of the above theories. Moreover, if Π and Π' are not classically equivalent but faithfully equivalent, they both have $Cn(\{x, y\})$ as their only extension.

Let us assume that a counterexample to unitary redundancy exists. There exists Π and Π' such that $\Pi' \subset \Pi$ and Π' is equivalent to Π but for every $\gamma \in \Pi$ it holds that $\Pi \backslash \{\gamma\}$ is not equivalent to Π. Since Π' is equivalent to Π, it cannot be the case that Π' and Π differ for a single clause γ, as otherwise Π' would be equal to $\Pi \backslash \{\gamma\}$ and therefore not equivalent to Π.

As a result, Π' and Π differ on two or more clauses, that is, $\Pi\backslash\Pi'$ contains at least two clauses. We show that, among the sets $\Pi\backslash\{\gamma\}$ such that $\gamma \in \Pi\backslash\Pi'$, there are at least two which are not classically equivalent. Let assume to the contrary that all these theories $\Pi\backslash\{\gamma\}$ are classically equivalent. Then, their union would be classically equivalent to them, and therefore also faithfully equivalent to them; since their union is Π, that would contradict the assumption that they are not equivalent to it.

We are therefore in the following condition: there are at least two clauses $\gamma_1, \gamma_2 \in \Pi$ such that no two theories among Π, $\Pi\backslash\{\gamma_1\}$, and $\Pi\backslash\{\gamma_2\}$ are equivalent to each other. Let $\Pi_1 = \Pi\backslash\{\gamma_1\}$ and $\Pi_2 = \Pi\backslash\{\gamma_2\}$. We have that Π' and Π are equivalent, but are not classically equivalent to Π_1 and Π_2, which are also not classically equivalent to each other.

Since Π' and Π are faithfully but not classically equivalent, they both have $Cn(\{x, y\})$ as their only extension, since every other possible set of extensions is only generated by classically equivalent theories. Since Π_1 and Π_2 are not faithfully equivalent to them, they are equivalent to two of the theories above: each one either entails y but not x, or entails $\neg x \vee \neg y$.

If either Π_1 or Π_2 entail $\neg x \vee \neg y$, their union does the same. Therefore, Π would not have $Cn(\{x, y\})$ as its only extension. If both theories entail y but not x, they are equivalent to y and $\neg x \wedge y$, respectively; their union is therefore $\neg x \wedge y$, which does not have $Cn(\{x, y\})$ as its only extension.

In all possible cases where unitary redundancy would be violated we have that reached a contradiction; therefore, no counterexample to unitary redundancy exists for this logic. □

The second counterexample shows that there exists a logic where the alternative definition of redundancy in terms of entailment ($\Pi\backslash\{\gamma\} \models_L \gamma$) is not the same as the definition in terms of equivalence ($\Pi \equiv_L \Pi\backslash\{\gamma\}$).

Counterexample 2 *(proof). There exists two relations $\models_L$ and $\equiv_L$ such that $\models_L$ is reflexive, $\equiv_L$ is reflexive and symmetric, $\models_L$ and $\equiv_L$ are related by full entailment, but there exists Π and $\gamma \in \Pi$ such that $\Pi\backslash\{\gamma\} \models_L \gamma$ but $\Pi \not\equiv_L \Pi\backslash\{\gamma\}$.*

Proof. We use faithful entailment $\models^e_D$ and equivalence $\equiv^e_D$ for the following set of defaults.

$$D = \{d_1, d_2\}$$

where

$$d_1 = \frac{: a \wedge \neg b}{a \wedge \neg b}$$

$$d_2 = \frac{a : b}{b}$$

In $\langle D, \emptyset\rangle$, the only applicable default is d_1, which blocks the application of d_2. The only extension of this theory is therefore $Cn(\{a, \neg b\})$. In $\langle D, \{a\}\rangle$ we can still apply d_1, with the same result, but we can also apply d_2. The consequence of this default blocks the application of the first; therefore, we have a second extension $Cn(\{a, b\})$.

We therefore have that $\mathrm{Ext}_D(\emptyset) \subseteq \mathrm{Ext}_D(\{a\})$ and therefore $\emptyset \models^e_D \{a\}$. Yet, $\emptyset \not\equiv^e_D \{a\}$ because $\mathrm{Ext}_D(\{a\}) \not\subseteq \mathrm{Ext}_D(\emptyset)$. As a result, by taking $\Pi = \{a\}$ and $\gamma = a$, we have that $\Pi\backslash\{\gamma\} \models^e_D \gamma$ but $\Pi\backslash\{\gamma\} \not\equiv^e_D \Pi$. □

Appendix B. Complexity of CIRC-redundancy

The following theorem shows the complexity of the problem of redundancy of a single clause.

Theorem 8 *(proof). Checking the CIRC-redundancy of a clause in a formula is Π^p_2-complete, and remains hard if the formula is assumed to be satisfiable.*

Proof. Lemma 1 proves that the redundancy of a clause in a formula can be checked by solving a $\forall\exists$QBF (for all $M\ldots$ there exists $M'\ldots$), and is therefore in Π^p_2. Alternatively, Theorem 4 and Theorem 6 prove that a clause γ is redundant in Π if and only if $\Pi\backslash\{\gamma\} \models_M \gamma$; Eiter and Gottlob [13] proved that this problem is in Π^p_2.

We show hardness by reduction from a QBF formula $\forall X \exists Y.\Gamma$ to the problem of redundancy. We assume that Γ is satisfiable; this assumption does not change the hardness of the problem. Let $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, and $\Gamma = \{\delta_1, \ldots, \delta_m\}$. Let a and $\{p_1, \ldots, p_n\}$ be new variables. We show that $\forall X \exists Y.\Gamma$ is valid if and only if γ is CIRC-redundant in Π, where:

$$\Pi = \{x_i \vee p_i \mid 1 \leqslant i \leqslant n\} \cup \{\neg a \vee y_j \mid 1 \leqslant j \leqslant n\} \cup \{a \vee \delta_k \mid 1 \leqslant k \leqslant m\} \cup \{\gamma\}$$
$$\gamma = \neg a \vee \neg y_1 \vee \cdots \vee \neg y_n$$

This formula Π is satisfiable: it is satisfied by every model where all p_i's are true, a is false, and all variables x_i's and y_i's have the values that make Γ true (such an assignment exists because Γ has been assumed satisfiable).

By Lemma 1, γ is CIRC-redundant in Π if and only if each minimal model of $\Pi \backslash \{\gamma\} \cup \neg\gamma$ contains a model of Π. The following equivalences holds:

$$\Pi \equiv \{x_i \vee p_i \mid 1 \leqslant i \leqslant n\} \cup \{\neg a\} \cup \Gamma$$
$$\Pi \backslash \{\gamma\} \cup \neg\gamma \equiv \{x_i \vee p_i \mid 1 \leqslant i \leqslant n\} \cup \{a, y_1, \ldots, y_n\}$$

The first equivalence holds because $\{\neg a \vee y_j \mid 1 \leqslant j \leqslant n\} \cup \{\neg a \vee \neg y_1 \vee \cdots \vee \neg y_n\}$ is equivalent to $\neg a$, as can be checked by resolving upon each y_j in turn. The second equivalence holds because $\neg\gamma = \{a, y_1, \ldots, y_n\}$ and this set implies all clauses $\neg a \vee y_j$ and $a \vee \delta_k$.

Formula $\Pi \backslash \{\gamma\} \cup \neg\gamma$ has a minimal model for each truth evaluation ω_X over the variables x_i:

$$I_{\omega_X} = \omega_X \cup \{p_i \mid 1 \leqslant i \leqslant n, x_i \notin \omega_X\} \cup \{a\} \cup \{y_j \mid 1 \leqslant j \leqslant n\}$$

We show that the model I_{ω_X} contains a model of Π if and only if $\Gamma|_{\omega_X}$ is satisfiable. By Lemma 1, the redundancy of γ corresponds to this condition being true for all possible models of $\Pi \backslash \{\gamma\} \cup \neg\gamma$. This would therefore prove that the QBF is valid if and only if γ is redundant in Π.

Since $\Pi \equiv \{x_i \vee p_i \mid 1 \leqslant i \leqslant n\} \cup \{\neg a\} \cup \Gamma$, if Γ has a model with a given value of ω_X then Π has a model that is strictly contained in I_{ω_X}: add to the satisfying assignment of Γ the setting of every p_i to the opposite of x_i and a to false.

On the converse, if Π has a model that is strictly contained in I_{ω_X}, this model must have exactly the same value of $X \cup P$ because Π contains $x_i \vee p_i$ and either x_i or p_i is false in I_{ω_X}. On the other hand, this model of Π must also set a to false and satisfy Γ, thus showing that there exists an assignment extending ω_X and satisfying Γ. $\square$

The following lemma is about making clauses irredundant while not changing the redundancy status of the other ones. In other words, it shows a way to make the clauses of $\Pi \backslash \Pi'$ *necessary*, that is, contained in all equivalent subsets of Π.

Lemma 4 *(proof). For every consistent formula $\Pi = \{\gamma_1, \ldots, \gamma_m\}$ and every $\Pi' \subseteq \Pi$, the only* CIRC*-redundant clauses of $I(\Pi, \Pi')$ are the clauses $\neg s \vee \neg t \vee \gamma_i$ such that $\gamma_i \in \Pi'$ is* CIRC*-redundant in Π.*

Proof. There are four possible assignment to the variables s and t. Since the models of $I(\Pi, \Pi')$ can be partitioned into the models of $I(\Pi, \Pi') \cup \{\neg s, \neg t\}$, $I(\Pi, \Pi') \cup \{s, \neg t\}$, $I(\Pi, \Pi') \cup \{\neg s, t\}$, and $I(\Pi, \Pi') \cup \{s, t\}$, the minimal models of $I(\Pi, \Pi')$ are necessarily some of the minimal models of these formulae.

In the table below we show what remains of $I(\Pi, \Pi') \backslash \{s \vee t\}$ in each of the four possible assignment to s and t after removing entailed clauses and false literals. We also show the minimal models of the resulting formulae.

assignment	subformula	minimal models
$\{\neg s, \neg t\}$	$\{a, b\}$	$\{a, b\}$
$\{s, \neg t\}$	$\{b\} \cup \{c_i \vee d_i \mid 1 \leqslant i \leqslant m\} \cup$ $\{\neg c_i \mid 1 \leqslant i \leqslant m\}$	$\{s, b\} \cup \{d_i\}$
$\{\neg s, t\}$	$\{a\} \cup \{c_i \vee \gamma_i \mid 1 \leqslant i \leqslant m, \gamma_i \in \Pi \backslash \Pi'\} \cup$ $\{x \vee x' \mid x \in Var(\Pi)\}$	$\{t, a\} \cup H$ where $H \subseteq C \cup X \cup X'$
$\{s, t\}$	$\{\neg c_i \mid 1 \leqslant i \leqslant m\} \cup$ $\{c_i \vee \gamma_i \mid 1 \leqslant i \leqslant m, \gamma_i \in \Pi \backslash \Pi'\} \cup \Pi'$	$\{s, t\} \cup H$ where $H \in \mathrm{CIRC}(\Pi)$

The four subformulae are all satisfiable. Moreover, no minimal model of one is contained in the minimal models of another because of either the values of $\{s, t\}$ and $\{a, b\}$. As a result, the minimal models of $I(\Pi, \Pi') \backslash \{s \vee t\}$ are exactly the minimal models of the four subformulae. The clause $s \vee t$ is irredundant because its addition to $I(\Pi, \Pi') \backslash \{s \vee t\}$ deletes the minimal model $\{a, b\}$. The minimal models of $I(\Pi, \Pi')$ are therefore exactly the minimal models of the remaining three subformulae.

We show that the remaining clauses but the ones derived from Π' are irredundant. This is shown by removing a clause from the set and showing that some of the minimal models of a subformula can be removed some elements. Since the minimal models of these three subformulae are exactly the minimal models of Π, this is a proof that the clause is irredundant.

1. The clauses $s \vee a$ and $t \vee b$ are irredundant because their removal would allow a and b to be set to false in the minimal models of the third and second subformula, respectively.
2. The clauses $\neg s \vee t \vee c_i \vee d_i$ and $\neg s \vee \neg c_i$ are irredundant because their removal would allow d_i to be set to false in the minimal model of the second subformula.
3. The clauses $\neg t \vee c_i \vee \gamma_i$ and $s \vee \neg t \vee x \vee x'$ require a longer analysis. In the third assignment, $I(\Pi, \Pi')$ becomes:

   $$F = \{a\} \cup \{c_i \vee \gamma_i \mid 1 \leqslant i \leqslant m\} \cup \{x \vee x' \mid x \in Var(\Pi)\}$$

   The clauses $x \vee x'$ are positive. By Lemma 2, they are CIRC-redundant if and only if they are redundant. In turn, they are not redundant because $\{a\} \cup \{c_i \mid 1 \leqslant i \leqslant m\} \cup \{y \mid y \in X,\ y \neq x\}$ is a model of all clauses but $x \vee x'$. Since c_i occurs positive in $c_i \vee \gamma_i$, Lemma 3 ensures that this clause is CIRC-redundant in F if and only if it is redundant in $F \backslash \{x \vee x' \mid \neg x \in \gamma_i\}$. This is false because the removal of $c_i \vee \gamma_i$ creates the following new model:

   $$M = \{c_j \mid 1 \leqslant j \leqslant m,\ j \neq i\} \cup \{x'\} \cup \{x \mid \neg x \in \gamma_i\}$$

   This model M satisfies $F \backslash \{x \vee x' \mid \neg x \in \gamma_i\} \backslash \{c_i \vee \gamma_i\}$: all clauses $c_j \vee \gamma_j$ are satisfied because $c_j \in M$ and all clauses $x \vee x'$ are satisfied because $x' \in M$. On the other hand, M does not satisfy $c_i \vee \gamma_i$ because it assigns all its literals to false.

The only clauses that can therefore be redundant are those corresponding to the clauses of Π'. In particular, these clauses only occur in the fourth subformula, which is equivalent to $\{s, t\} \cup \{\neg c_i \mid 1 \leqslant i \leqslant m\} \cup \Pi$. A clause $\neg s \vee \neg t \vee \gamma_i$ with $\gamma_i \in \Pi'$ is therefore CIRC-redundant in $I(\Pi, \Pi')$ if and only if γ_i is CIRC-redundant in Π. $\square$

Appendix C. Equivalence in default logics

In this section we show the relationship among the different definitions of equivalence in default logic.

C.1. Non-equality of equivalence relations

In this section we show that the three considered forms of equivalence do not in general coincide even if one theory is contained in the other.

Counterexample 9. *There exists a set of defaults D and two formulae W and W' such that $W' \subset W$, $W' \equiv^m_D W$, and $W' \not\equiv^c_D W$ in Reiter, justified, constrained and rational default logic.*

Proof. Let D, W, and W' be as follows.

$$D = \{d_1, d_2\}$$

where:

$$d_1 = \frac{: a \wedge \neg b}{a \wedge \neg b}$$

$$d_2 = \frac{a : b}{b}$$

$$W = \{a\}$$

$$W' = \emptyset$$

The only selected process of $\langle D, W'\rangle$ is $[d_1]$, which generates the extension $Cn(\{a, \neg b\})$. This extension entails W, but it also entails $\neg b$. The theory $\langle D, W\rangle$ has also the process $[d_2]$, generating the extension $Cn(\{a, b\})$. These two processes cannot however be concatenated, as the consequence $\neg b$ of d_1 is inconsistent with the justification of d_2.

Since $W' \subset W$, we have that $W \models_D W'$. In this example, we also have $W' \models_D W$ because the single extension of W' entails $W = \{a\}$. However, W' and W have different set of extensions; in particular, $\vee\mathrm{Ext}_D(W') \equiv a \wedge \neg b$ and $\vee\mathrm{Ext}_D(W) \equiv a$. □

Since $W' \not\equiv^c_D W$, it also holds $W' \not\equiv^e_D W$, which proves that mutual and faithful equivalence do not coincide. A similar result can be proved about $\equiv^c_D$ and $\equiv^e_D$.

Counterexample 10. *There exists a set of defaults D and two formulae W and W' such that $W' \subset W$, $W' \equiv^c_D W$, and $W' \not\equiv^e_D W$ in Reiter and justified default logic.*

Proof. Let $D = \{d_1, d_2, d_3, d_4\}$, $W = \{a\}$, $W' = \emptyset$, and the defaults be as follows.

$$d_1 = \frac{: \neg b \wedge c}{a \wedge c}$$
$$d_2 = \frac{: \neg b \wedge \neg c}{a \wedge \neg c}$$
$$d_3 = \frac{a : b \wedge c}{b \vee c}$$
$$d_4 = \frac{a \wedge (b \vee c) : b \wedge \neg c}{b}$$

In W', only d_1 and d_2 are applicable. The consequence of the first default is $a \wedge c$, which is consistent with the justification of d_3. Default d_4 is not applicable because of its justification. The first selected process of W' is therefore $[d_1, d_3]$, leading to the extension $Cn(\{a, c\})$.

The second process from W' starts with d_2, whose consequence is $a \wedge \neg c$, which is not consistent with d_1 and d_3, and does not imply the precondition of d_4. As a result, $[d_2]$ is the second selected process of W', leading to the extension $Cn(\{a, \neg c\})$. We therefore have $\vee\mathrm{Ext}_D(W') \equiv (a \wedge c) \vee (a \wedge \neg c) \equiv a$.

Let us now consider the extensions of W. All selected processes of W' are also selected processes of W. However, we can now apply d_3, as a is true in the background theory. We therefore obtain $b \vee c$. This conclusion is inconsistent with the justification of d_2, but d_1 and d_4 can be applied. The first one leads to the extension $Cn(\{a, c\})$, which is also an extension of $\langle D, W'\rangle$. On the other hand, $[d_3, d_4]$ leads to $Cn(\{a, b\})$, which is a new extension. Nevertheless, $\vee\mathrm{Ext}_D(W) \equiv (a \wedge c) \vee (a \wedge \neg c) \vee (a \wedge b) \equiv a$: the theory $\langle D, W\rangle$ has the extensions $Cn(\{a, b\})$ which $\langle D, W'\rangle$ does not have, but their skeptical consequences are the same. □

In the proof, we used two defaults that are applicable in W but not in the processes of $\langle D, W'\rangle$. Such two defaults cannot have mutually consistent justifications; otherwise, they would be both applicable in some process of $\langle D, W'\rangle$ thanks to the fact that any extension of $\langle D, W\rangle$ contains only models of some extensions of $\langle D, W'\rangle$. This condition is not possible in constrained and rational default logic; however, the same claim can be proved in a different way.

Counterexample 11. *There exists a set of defaults D and two formulae W and W' such that $W' \subset W$, $W' \equiv^c_D W$, and $W' \not\equiv^e_D W$ in constrained and rational default logic.*

Proof. D, W, and W' are defined as follows.

$$D = \{d_1, d_2, d_3, d_4\}$$
$$W = \{x\}$$
$$W' = \emptyset$$

$$d_1 = \frac{: x \wedge a}{x}$$
$$d_2 = \frac{x : a \wedge b}{b}$$
$$d_3 = \frac{x : a \wedge \neg b}{\neg b}$$
$$d_4 = \frac{x : \neg a \wedge \neg b}{x}$$

The justifications of d_2, d_3, and d_4 are mutually inconsistent. The selected processes of $\langle D, W\rangle$ are $[d_1, d_2]$, $[d_2, d_1]$, $[d_1, d_3]$, $[d_3, d_1]$, and $[d_4]$, generating the three extensions $E_1 = Cn(\{x, b\})$, $E_2 = Cn(\{x, \neg b\})$, and $E_3 = Cn(\{x\})$. Their disjunction is $\vee \mathrm{Ext}_D(W) \equiv x$.

Let us now consider $\langle D, W'\rangle$. The only default that is applicable in $W' = \emptyset$ is d_1, which generates x but also has a as a justification. As a result, the defaults d_2 and d_3 are still applicable, but d_4 is not. As a result, the only extensions of $\langle D, W'\rangle$ are $Cn(\{x, b\})$ and $Cn(\{x, \neg b\})$. We therefore have $W' \not\equiv^e_D W$. On the other hand, $\vee \mathrm{Ext}_D(W') \equiv x$, which is equivalent to $\vee \mathrm{Ext}_D(W)$. As a result, $W' \equiv^c_D W$. □

C.2. Equality of entailment relations

The following chain of implications is easy to prove:

$$W' \models^e_D W \Rightarrow W' \models^c_D W \Rightarrow W' \models_D W$$

The latter implication is proved by the following lemma.

Lemma 7. *For any set of defaults D and two formulae W and W', if $W' \models^c_D W$ then $W' \models_D W$, in Reiter, justified, constrained and rational default logic.*

Proof. By assumption, $\vee \mathrm{Ext}_D(W') \models \vee \mathrm{Ext}_D(W)$. Since every extension of $\langle D, W\rangle$ entails W, we have $\vee \mathrm{Ext}_D(W) \models W$. As a result, $\vee \mathrm{Ext}_D(W') \models W$, which is by definition $W' \models_D W$. □

Redundancy is defined in terms of equivalence of two formulae, one contained in the other. As a result, it makes sense to study the conditions of equivalence in the particular case in which $W' \subseteq W$. The above chain of implications can be wrapped around in this case, thus proving that the three forms of entailment coincide. The following lemma has been proved by Makinson [35, Observation 9] for Reiter default logic; Makinson stated that if $W' \subseteq W$ and $W' \models_D W$ then $W' \models^c_D W$, but the proof of this statement actually shows that $W' \models^c_D W$ holds by proving that $W' \models^e_D W$ holds in this case. This fact can be generalized to justified, constrained, and rational default logic.

Lemma 8. *For any set of defaults D and two formulae W and W', if $W' \subseteq W$ and $W' \models_D W$ then $W' \models^e_D W$ for Reiter, justified, constrained and rational default logic.*

Proof. Let Π be a selected process of $\langle D, W'\rangle$. We prove that it is also a selected process of $\langle D, W\rangle$. Since $W' \models_D W$, the formula W is entailed by every extension in $\mathrm{Ext}_D(W')$. In particular, $W' \cup \mathrm{cons}(\Pi) \models W$. Therefore, $W' \cup \mathrm{cons}(\Pi) \equiv W \cup \mathrm{cons}(\Pi)$. As a result, all conditions (such as success and closure) where W' only occurs in the subformula $W' \cup \mathrm{cons}(\Pi)$ are not changed by replacing W' with W. This is in particular true for all considered conditions of success and closure.

The only condition that mentions the background theory not in conjunction with $\mathrm{cons}(\Pi)$ is the condition of a sequence being a process: Π is a process of $\langle D, W'\rangle$ if and only if $W' \cup \mathrm{cons}(\Pi[d]) \models \mathrm{prec}(d)$ for any $d \in \Pi$. The same condition is however true for W because $W' \subseteq W$ implies $W \models W'$. □

The following is a consequence of these two lemmas.

Corollary 4. *For any set of defaults D and two formulae W and W', if $W' \subseteq W$, then in Reiter, justified, constrained, and rational default logic the following holds*:

$$W' \models_D W \Leftrightarrow W' \models^c_D W \Leftrightarrow W' \models^e_D W$$

C.3. Equality of equivalence relations in particular cases

While $\equiv^c_D$ and $\equiv^e_D$ are not the same in general, they coincide when all defaults are normal and one formula is contained in the other one.

Theorem 16. *For any set of normal defaults D and two formulae W and W', if $W' \subseteq W$ then $W' \equiv^c_D W$ implies $W' \equiv^e_D W$ in constrained default logic.*

Proof. Given the previous result, we only have to prove that $W \equiv^c_D W'$ implies that $\mathrm{Ext}_D(W) \subseteq \mathrm{Ext}_D(W')$, that is, $\langle D, W\rangle$ does not have any extension that is not an extension of $\langle D, W'\rangle$.

To the contrary, assume that such extension exists. Let Π be the process that generates the extension of $\langle D, W\rangle$ that is not an extension of $\langle D, W'\rangle$. By definition of process, $\mathrm{cons}(\Pi) \cup W \cup \mathrm{just}(\Pi)$ is consistent. Therefore, it has a model M. Since this model satisfies both W and $\mathrm{cons}(\Pi)$, it is a model of the extension generated by Π.

Since the conclusions of the two theories are the same, every model of the extension generated by Π is a model of some extensions of $\langle D, W'\rangle$. Let Π' be the process of $\langle D, W'\rangle$ that generates an extension that has the model M. We prove that all defaults of Π are in Π'.

Since M is a model of the extension generated by Π', it is a model of $\mathrm{cons}(\Pi') \cup W'$. Therefore, it is a model of $\mathrm{cons}(\Pi')$, and a model of $\mathrm{just}(\Pi')$ because defaults are normal. We have already proved that M is a model of $\mathrm{cons}(\Pi)$ and $\mathrm{just}(\Pi)$ and of W. As a result, the set $\mathrm{cons}(\Pi) \cup \mathrm{cons}(\Pi') \cup W \cup \mathrm{just}(\Pi) \cup \mathrm{just}(\Pi')$ is consistent. Therefore, we can add all defaults of Π to Π' without contradicting the justifications.

As a result, the defaults of Π are not in Π' only if their preconditions are not entailed from the consequences of Π. This is impossible: since Π is a process of $\langle D, W\rangle$, we have $W \models \mathrm{prec}(d)$, where d is the first default of Π. As a result, d must be part of Π', otherwise Π' would not be a maximal process. The consequences of d are therefore part of $\mathrm{cons}(\Pi') \cup W'$. Repeating the argument with the second default of Π we get the same result. We can therefore conclude that all defaults of Π are in Π'. $\square$

Since Reiter, justified, constrained, and rational default logics coincide on normal default theories, the equality of the definitions of equivalence holds when defaults are normal.

Theorem 17. *If D is a set of normal defaults and W and W' are two formulae such that $W' \subseteq W$, then $W' \equiv^c_D W$ holds if and only if $W' \equiv^e_D W$ holds for Reiter, justified, constrained, and rational default logic.*

The following lemma is what is missing for proving that the three considered forms of equivalence coincide when all defaults are categorical and normal.

Lemma 9. *If D is a set of categorical defaults and W and W' are two formulae such that $W' \subseteq W$ and $W' \models_D W$, then $W \models^e_D W'$ in constrained default logic.*

Proof. Let Π be a selected process of $\langle D, W\rangle$. We prove that Π is a selected process of $\langle D, W'\rangle$ generating the same extension.

Since Π is a selected process of $\langle D, W\rangle$, it holds that $W \cup \mathrm{cons}(\Pi) \cup \mathrm{just}(\Pi)$ is consistent. Since $W' \subseteq W$, it also holds that $W' \cup \mathrm{cons}(\Pi) \cup \mathrm{just}(\Pi)$ is consistent. Since no default has preconditions, Π is a successful process of $\langle D, W'\rangle$. Since constrained default logic is a failsafe semantics [29], there exists Π' such that $\Pi \cdot \Pi'$ is a selected process of $\langle D, W\rangle$.

Since every extension of W' entails W, this is in particular true for the extension generated by $\Pi \cdot \Pi'$. In other words, $W' \cup \mathrm{cons}(\Pi \cdot \Pi') \models W$. As a result, $W' \cup \mathrm{cons}(\Pi \cdot \Pi') \equiv W \cup \mathrm{cons}(\Pi \cdot \Pi')$. Since $\Pi \cdot \Pi'$ is a process of $\langle D, W'\rangle$, we have that $W' \cup \mathrm{cons}(\Pi \cdot \Pi') \cup \mathrm{just}(\Pi \cdot \Pi')$ is consistent, which is therefore equivalent to the consistency of $W \cup \mathrm{cons}(\Pi \cdot \Pi') \cup \mathrm{just}(\Pi \cdot \Pi')$. Therefore, $\Pi \cdot \Pi'$ is a successful process of $\langle D, W\rangle$. Since Π is by assumption a maximal successful process of $\langle D, W\rangle$, it must be the case that $\Pi' = [\,]$, that is, $\Pi \cdot \Pi' = \Pi$. We have already proved that $W' \cup \mathrm{cons}(\Pi \cdot \Pi') \equiv W \cup \mathrm{cons}(\Pi \cdot \Pi')$, that is, Π generates the same extension in W and in W'. $\square$

Since constrained and Reiter default logics coincide on normal default theories, we have the following consequence.

Corollary 5. *If D is a set of normal and categorical defaults and W and W' are two formulae such that $W' \subseteq W$, the conditions $W' \equiv_D^m W$, $W' \equiv_D^c W$, and $W' \equiv_D^e W$ are equivalent in Reiter, justified, constrained, and rational default logic.*

Appendix D. Complexity of redundancy in default logic

D.1. Complexity of clause redundancy

In this section, we analyze the complexity of checking the redundancy of a clause in a formula. Formally, this is the problem of whether $W \backslash \{\gamma\}$ is equivalent to W according to $\equiv_D^c$ or $\equiv_D^e$. By Corollary 4, these two forms of equivalence are related, as $W' \models_D^c W$ is equivalent to $W' \models_D^e W$ (and also to $W' \models_D W$), if $W' \subseteq W$. As a result, checking whether $W' \models_D W$ allows for telling whether the "first part of equivalence" between W' and W holds, for both kinds of equivalence. In other words, in order to check whether W' and W are equivalent, if $W' \subseteq W$ we can first check whether $W' \models_D W$; if this condition is true, we then proceed checking whether $W \models_D^c W'$ or $W \models_D^e W'$ depending on which equivalence is considered.

Lemma 8 tells that $W' \models_D W$ implies that all extensions of $\langle D, W' \rangle$ are also extensions of $\langle D, W \rangle$. This condition does not imply equivalence because $\langle D, W \rangle$ may contain some other extension, as in the default theory $\langle D, W \rangle$ below.

$$W = \{a\}$$
$$D = \{d_1, d_2\}$$

where:

$$d_1 = \frac{a : b}{b}$$
$$d_2 = \frac{: a \wedge \neg b}{a \wedge \neg b}$$

The theory $\langle D, W \rangle$ has two extensions: applying either d_1 or d_2, the other is not applicable. The resulting extensions are $Cn(\{a, b\})$ and $Cn(\{a, \neg b\})$. Let $W' = \emptyset$. The only default that is applicable in W' is d_2, leading to the only extension $Cn(\{a, \neg b\})$. This extension implies W. As a result, we have that $W' \models_D W$ but W' and W do not have the same extensions and the same consequences. In particular, W has some extensions that W' does not have. This is always the case if $W' \models_D W$ but W and W' are not equivalent.

In order to check equivalence of W' and W with $W' \subseteq W$, two conditions have to be checked:

1. $W' \models_D W$; and
2. $W \models_D^c W'$ or $W \models_D^e W'$.

An upper bound on the complexity of checking the redundancy of a clause is given by the following theorem.

Theorem 18. *Checking whether $W' \equiv_D^e W$ for a set of defaults D and two formulae W and W' such that $W' \subseteq W$ is in Π_2^p for Reiter and justified default logic and in Π_3^p for constrained and rational default logic.*

Proof. Checking whether $W' \models_D W$ is in Π_2^p. The other condition to be checked is $W \models_D^e W'$. The converse of this condition is that there exists a formula $E \subseteq W \cup \text{cons}(D)$ such that $Cn(E)$ is an extension of $\langle D, W \rangle$ but is not an extension of $\langle D, W' \rangle$. Since checking whether a formula is a Reiter or justified default logic is in $\Delta_2^p[\log n]$ [28,45], the whole problem is in Σ_2^p. Its converse is therefore in Π_2^p. The problem of redundancy of a clause can be solved by solving two problems in Π_2^p in parallel.

The same line of proof does not work for constrained and rational default logic, where extension checking is Σ_2^p-complete [28]; this makes clause redundancy to be in Π_3^p for these two semantics. □

Clause redundancy is Π_2^p-hard for all considered semantics, as shown by the following theorem.

Theorem 19. *Checking whether $W' \equiv_D^e W$, where D is a set of defaults and W and W' are two formulae, is Π_2^p-hard even if $W = W' \cup \{a\}$ where a is a variable, W is consistent, and all defaults are categorical and normal, in Reiter, justified, constrained, and rational default logic.*

Proof. The claim could be proved from the fact that entailment in default logic is Π_2^p-hard even if the formula to entail is a single positive literal, and all defaults are categorical and normal [20,49]. If all defaults are categorical and normal, Corollary 5 proves that $W' \equiv_D^m W$ is equivalent to the two other forms of equivalence.

We show a new reduction from ∀∃QBF. The formula $\forall X \exists Y \,.\, F$ is valid if and only if the new variable a is redundant in the theory below:

$$\left\langle \left\{ \frac{:x_i}{x_i}, \frac{:\neg x_i}{\neg x_i} \right\} \cup \left\{ \frac{:F \wedge a}{a} \right\}, \{a\} \right\rangle$$

The background theory $W = \{a\}$ is consistent. This theory has an extension for every possible truth assignment over the variables X. For each such extension, the last default can be applied only if F is consistent with the given assignment over X. As a result, if F is consistent with every truth assignment over the variables X, then a can be removed from the background theory without changing the consequences of these extensions. Otherwise, the removal of a would cause some of these extensions not to entail a any more. □

The following theorem shows that the same problem is Π_3^p-complete for the other two considered semantics (constrained and rational).

Theorem 20. *Checking whether $W \equiv_D^e W'$, where D is a set of defaults and W and W' two formulae, is Π_3^p-hard for constrained and rational default logic even if $W = W' \cup \{a\}$, where a is a variable.*

Proof. We show a reduction from ∀∃∀QBF to the problem under consideration. Given a formula $\forall X \exists Y \forall Z.F$, where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$, the corresponding redundancy problem is defined by $W = \{a\}$, $W' = \emptyset$, and $D = D_1 \cup D_2 \cup D_3 \cup D_4$; these sets of defaults are defined as follows, where a, b, and f are new variables, and $C = \{c_1, \ldots, c_n\}$ and $E = \{e_1, \ldots, e_n\}$ are sets of new variables.

$$D_1 = \left\{ \frac{a : \neg b \wedge x_i}{x_i}, \frac{a : \neg b \wedge \neg x_i}{\neg x_i} \,\middle|\, 1 \leqslant i \leqslant n \right\}$$

$$D_2 = \left\{ \frac{a : \neg b}{C \wedge \neg d \wedge E \wedge f} \right\}$$

$$D_3 = \left\{ \frac{: b \wedge x_i}{c_i \wedge (d \vee x_i)}, \frac{: b \wedge \neg x_i}{c_i \wedge (d \vee \neg x_i)} \,\middle|\, 1 \leqslant i \leqslant n \right\}$$

$$D_4 = \left\{ \frac{C : b \wedge y_i}{e_i \wedge (f \vee y_i)}, \frac{C : b \wedge \neg y_i}{e_i \wedge (f \vee \neg y_i)} \,\middle|\, 1 \leqslant i \leqslant n \right\}$$

$$D_5 = \left\{ \frac{C \wedge E \wedge (d \vee f \vee F) : b}{a \wedge \neg d \wedge f} \right\}$$

The defaults in $D_1 \cup D_2$ all have $\neg b$ as a justification, while the defaults in $D_3 \cup D_4 \cup D_5$ all have b. As a result, any successful process either contains only defaults from the first set or only defaults from the second set.

Since the defaults of the first set all have a as a precondition, they are applicable from W, but they are not applicable from W'. The default in D_5 is the only default having a as a consequence, and this default has b as a justification; as a result, even if a could be derived from W', this requires the application of the default in D_5, which blocks the defaults in $D_1 \cup D_2$.

We can therefore conclude that every selected process of $\langle D, W\rangle$ is made either of defaults of $D_1 \cup D_2$ or of defaults of $D_3 \cup D_4 \cup D_5$, while the selected processes of $\langle D, W'\rangle$ are all made of defaults of the second set.

We can also show that every selected process of W' is also a selected process of W. Let Π be a selected process of W'. Since a never occurs negated, the consistency of $W' \cup \mathrm{cons}(\Pi) \cup \mathrm{just}(\Pi)$ entails that of $W \cup \mathrm{cons}(\Pi) \cup \mathrm{just}(\Pi)$. As for the condition of maximality, Π contains at least a default of $D_3 \cup D_4 \cup D_5$ (e.g., $\frac{:b \wedge x_1}{c_1 \wedge (d \vee x_1)}$ is applicable from W')

and therefore has b as a justification. As a result, the defaults in $D_1 \cup D_2$ cannot be applied because they all have $\neg b$ in their justifications. Therefore, Π is also maximal in W.

The only way W and W' may not have the same extensions is when W has an extension made of defaults of $D_1 \cup D_2$ such that W' does not generate the same extension using the defaults of $D_3 \cup D_4 \cup D_5$. The extensions of W generated from the extensions of $D_1 \cup D_2$ are in bijective correspondence with the interpretations over the variables X: for every such interpretation ω_X there is an extension that is equivalent to $a \wedge C \wedge \neg d \wedge E \wedge f \wedge \omega_X$, and vice versa. We now show how W' could possibly generate such an extension using the defaults in $D_3 \cup D_4 \cup D_5$.

The only defaults that are applicable in W' are those in D_3. Since the set of all justifications has to be consistent, two defaults containing opposite values of x_i cannot be applied. A maximal such process has consequences equivalent to $C \wedge (d \vee \omega_X)$ where ω_X is an interpretation over the variables X. More precisely, for every interpretation there are some such processes, and vice versa.

Since the consequences of this process entail C, the defaults of D_4 can now be applied. Again, what results is a process whose consequences also include $E \wedge (f \vee \omega_Y)$. In other words, for every pair of interpretations ω_X and ω_Y over X and Y, respectively, there are some processes all having consequences equivalent to $C \wedge (d \vee \omega_X) \wedge E \wedge (f \vee \omega_Y)$, and vice versa.

The default in D_5 now could be applied from this process; more precisely, $C \wedge E$ is now entailed by the consequence of this process; the default is applicable if and only if $d \vee f \vee F$ is also entailed. Since d and f are new variables, this is equivalent to $\omega_X \wedge \omega_Y \models F$, that is, F is valid for any value of Z in the interpretation $\omega_X \cup \omega_Y$.

The effect of the application of this default is that d is made false while a and f are made true. The resulting extension $a \wedge \neg d \wedge f \wedge C \wedge (d \vee \omega_X) \wedge E \wedge (f \vee \omega_Y)$ is equivalent to $a \wedge C \wedge \neg d \wedge E \wedge f \wedge \omega_X$. This extension is always generated by W from defaults in $D_1 \cup D_2$, while W' generates it only if the default in D_5 can be applied.

Now, assume that $\forall X \exists Y \forall Z . F$ is valid. This means that for every extension equivalent to $a \wedge C \wedge \neg d \wedge E \wedge f \wedge \omega_X$ of W, there exists an interpretation ω_Y such that the process that has $C \wedge (d \vee \omega_X) \wedge E \wedge (f \vee \omega_Y)$ as a consequence can be extended by the application of the default of D_5, thus generating the same extension.

If instead $\forall X \exists Y \forall Z . F$ is not valid, there exists ω_X such that, for every ω_Y, it holds $\omega_X \wedge \omega_Y \not\models F$. In terms of extensions of the two considered theories, W still has the extension equivalent to $a \wedge C \wedge \neg d \wedge E \wedge f \wedge \omega_X$. However, the only processes of W' that can generate the same extension are the ones that extend the process having $C \wedge (d \vee \omega_X) \wedge E \wedge (f \vee \omega_Y)$ for some ω_Y has consequence. By assumption, F is not entailed by $\omega_X \cup \omega_Y$; therefore, the default of D_5 cannot be applied in any such process. This means that the sets of extensions of W and W' differ. □

We now consider the problem of redundancy of clauses when consequence-equivalence is used. The difference between the two kinds of equivalence is that two sets of extensions may be different but yet the disjunctions of their elements are the same. The necessity of calculating the disjunction of all extensions intuitively explains why checking redundancy for consequence-equivalence is harder than for faithful equivalence.

Theorem 21. *Checking whether $W' \equiv_D^c W$, where D is a set of defaults and W and W' are two formulae, is in Π_3^p if $W' \subseteq W$ for Reiter, justified, constrained, and rational default logic.*

Proof. W' and W are consequence-equivalent if $W' \models_D W$ and $W \models_D^c W'$. The first problem is in Π_2^p. We prove that the converse of the second condition is in Σ_3^p. By definition, $W \not\models_D^c W'$ holds if and only if $\vee \mathrm{Ext}_D(W) \not\models \vee \mathrm{Ext}_D(W')$. In terms of models, we have $\cup \{Mod(E) \mid E \in \mathrm{Ext}_D(W)\} \not\subseteq \cup \{Mod(E) \mid E \in \mathrm{Ext}_D(W')\}$, that is, there exists M and E such that $M \in Mod(E)$, $E \in \mathrm{Ext}_D(W)$, but M is not a model of any extension of W'. The whole condition can therefore be expressed by the following formula.

$$\exists M \exists E \, . \, M \in Mod(E) \wedge E \in \mathrm{Ext}_D(W) \wedge (\forall E' \, . \, E' \notin \mathrm{Ext}_D(W') \vee M \notin Mod(E'))$$

Since $E' \notin \mathrm{Ext}_D(W')$ is in $\Delta_2^p[\log n]$ for Reiter [45] and justified default logic and in Π_2^p for constrained and rational [28], the problem of checking $W \models_D^c W'$ is in Π_3^p. Therefore, the problem of consequence-equivalence is in Π_3^p as well for all four considered semantics. □

We show that the problem is hard for the same class.

Theorem 22. *The problem of checking whether $W' \equiv^c_D W$, where D is a set of default and W and W' two formulae, is Π^p_3-hard even if $W = W' \cup \{a\}$ where a is a variable for Reiter justified default, constrained and rational default logics.*

Proof. We prove the claim by reduction from QBF. We reduce a formula $\exists X \forall Y \exists Z.F$ into the problem of checking whether $W \not\equiv^c_D W'$, where $W' = \emptyset$, $W = \{a\}$, and $D = D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6$. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$. We show each D_i at time. First, we generate a complete assignment over the variables X using the following defaults, where $H = \{h_1, \ldots, h_n\}$ are new variables.

$$D_1 = \left\{ \frac{: x_i}{x_i \wedge h_i}, \frac{: \neg x_i}{\neg x_i \wedge h_i} \right\}$$

Since these defaults have no preconditions, they can be applied regardless of whether W or W' is the background theory. They generate a process for any truth assignment ω_X over the variables in X. The variables h_i are all true only when all variables x_i have been set to a value.

The processes of W and W' are so far the same. Once all h_i are true, we can apply the defaults of $D_2 = \{d_1, d_2, d_3, d_4\}$, which are the ones used in Counterexample 10 to show two theories that have the same consequences but different extensions (a, b, and c are new variables):

$$\begin{aligned} D_2 &= \{d_1, d_2, d_3, d_4\} \\ d_1 &= \frac{h_1 \wedge \cdots \wedge h_n : \neg b \wedge c}{a \wedge c} \\ d_2 &= \frac{h_1 \wedge \cdots \wedge h_n : \neg b \wedge \neg c}{a \wedge \neg c} \\ d_3 &= \frac{h_1 \wedge \cdots \wedge h_n \wedge a : b \wedge c}{b \vee c} \\ d_4 &= \frac{h_1 \wedge \cdots \wedge h_n \wedge a \wedge (b \vee c) : b \wedge \neg c}{b} \end{aligned}$$

Since these defaults all have $\wedge H$ as a precondition, they can only be applied once a truth assignment over X has been generated by the previous defaults. They act as in the proof of Counterexample 10. Only $[d_1, d_3]$ and $[d_2]$ are processes of W'; their consequences are $a \wedge c$ and $a \wedge \neg c$. The theory W has the same processes, but also $[d_3, d_1]$ and $[d_3, d_4]$, which generate the extensions $a \wedge c$ and $a \wedge b$, respectively. While the first is also an extension of W', the second is not. The disjunction of all extensions is equivalent to $a \wedge H$ for both W and W'.

The idea is as follows: from $H \wedge a \wedge b \wedge \omega_X$, which is obtained from W but not from W', we always generate an extension equivalent to $H \wedge a \wedge b \wedge \omega_X \wedge d \wedge \epsilon_Y$, where ϵ_Y is the assignment of false to all variables of Y; from the two other partial extensions $H \wedge a \wedge \neg c \wedge \omega_X$ and $H \wedge a \wedge c \wedge \omega_X$ we instead generate an arbitrary assignment ω_Y, which then has $H \wedge a \wedge b \wedge \omega_X \wedge d \wedge \epsilon_Y$ as a model only if $\omega_X \wedge \omega_Y \wedge F$ is satisfiable.

This way, if there exists an assignment ω_X such that for all ω_Y the formula $\omega_X \wedge \omega_Y \wedge F$ is satisfiable, then there is no extension of W' having the model $H \cup \{a, b, d\} \cup \omega_X \cup \epsilon_Y$. Vice versa, if there exists even a single ω_Y such that $\omega_X \wedge \omega_Y \wedge F$ is unsatisfiable, an extension equivalent to $H \wedge a \wedge c \wedge \omega_X \wedge \ldots$ for W' will be generated, and this extension has the model $H \cup \{a, b\} \cup \omega_X \cup \{d\} \cup \epsilon_Y$.

The required defaults are the following ones. First, we generate the considered model from the process that generated $H \cup \{a, b\} \cup \omega_X$ by the following default, where d is a new variable:

$$D_3 = \left\{ \frac{b : \top}{d \wedge \neg y_1 \wedge \cdots \wedge \neg y_n} \right\}$$

From $H \wedge a \wedge \neg c \wedge \omega_X$ and $H \wedge a \wedge c \wedge \omega_X$ we generate an arbitrary truth assignment over Y. Since the model $H \cup \{a, b, d\} \cup \omega_X \cup \epsilon_Y$ assigns false to all variables y_i, we cannot simply add y_i as a conclusion. A similar effect can be achieved by the following defaults, where $L = \{l_1, \ldots, l_n\}$ are new variables.

$$D_4 = \left\{ \frac{\neg c : \neg d \wedge y_i}{d \vee (y_i \wedge l_i)}, \frac{\neg c : \neg d \wedge \neg y_i}{d \vee (\neg y_i \wedge l_i)} \;\middle|\; 1 \leqslant i \leqslant n \right\}$$

$$D_5 = \left\{ \frac{c : \neg d \wedge y_i}{d \vee (y_i \wedge l_i)}, \frac{c : \neg d \wedge \neg y_i}{d \vee (\neg y_i \wedge l_i)} \;\middle|\; 1 \leqslant i \leqslant n \right\}$$

The two defaults associated with y_i and $\neg y_i$ cannot be applied both at the same time, as the consequence of one contains the negation of the justification of the other one. Since the following defaults can only be applied when $d \vee (l_1 \wedge \cdots \wedge l_n)$ has been derived, the current extensions before their application are equivalent to $H \wedge a \wedge \neg c \wedge \omega_X \wedge (d \vee (\omega_Y \wedge L))$ and $H \wedge a \wedge c \wedge \omega_X \wedge (d \vee (\omega_Y \wedge L))$, where ω_Y is an arbitrary truth assignment over Y.

These extensions have all models of $a \wedge b \wedge \omega_X \wedge d \wedge \epsilon_Y$. The following default removes these models from the extensions if and only if $\omega_X \wedge \omega_Y \wedge F$ is satisfiable.

$$D_6 = \left\{ \frac{d \vee (l_1 \wedge \cdots \wedge l_n) : \neg d \wedge F}{\neg d} \right\}$$

This default is not applicable from $H \wedge a \wedge b \wedge \omega_X \wedge d \wedge \epsilon_Y$ because its justification contains $\neg d$. It is applicable from the other processes but only after the ith default of D_4 or D_5 has been applied for each i and only if the consequences of the applied defaults of D_4 or D_5 are consistent with $\neg d \wedge F$. In other words, $(d \vee (\omega_Y \wedge L)) \wedge \neg d \wedge F$ must be consistent, which is equivalent to the consistency of $\omega_Y \wedge F$ because d and L are not mentioned in ω_Y and F.

We can therefore conclude that:

1. for each truth assignment ω_X, three "partial extensions" are generated from W: $H \wedge a \wedge \neg c \wedge \omega_X$, $H \wedge a \wedge c \wedge \omega_X$, and $H \wedge a \wedge b \wedge \omega_X$; the first two are also generated by W';
2. from $H \wedge a \wedge b \wedge \omega_X$, an extension equivalent to $H \wedge a \wedge b \wedge \omega_X \wedge d \wedge \epsilon_Y$ is generated; if the models of this extension are not models of an extension of W', equivalence between W and W' does not hold;
3. from $H \wedge a \wedge [\neg]c \wedge \omega_X$ we generate $H \wedge a \wedge [\neg]c \wedge \omega_X \wedge (d \vee (\omega_Y \wedge L))$ for each truth assignment ω_Y on the variables Y; to this formula, $\neg d$ is added if and only if F is consistent with ω_X and ω_Y.

As a result, the models of $H \wedge a \wedge b \wedge \omega_X \wedge d \wedge \epsilon_Y$ are not models of an extension of W' if and only if $F \wedge \omega_X$ is satisfiable for every truth assignment over Y. Since non-equivalence has to be checked for every ω_X, we have that non-equivalence holds if and only if $\exists X \forall Y \exists Z \,.\, F$.

A similar proof holds for constrained or rational default logics by replacing the default theory of Counterexample 10 with that of Counterexample 11. The proof can also slightly simplified in this case, as the defaults of D_4 and D_5 can be modified with justifications y_i or $\neg y_i$ and consequence $d \vee l_i$. □

Since we have proved that the problem of clause redundancy w.r.t. consequence-equivalence is both in Π_3^p and hard for the same class, we have the following result.

Corollary 6. *The problem of checking whether $W' \equiv_D^c W$, where D is a set of default and W and W' two formulae, is Π_3^p-complete if $W' \subseteq W$; hardness holds even if $W = W' \cup \{a\}$ where a is a variable. This holds for Reiter, justified, constrained, and rational default logic.*

D.2. Complexity of formula redundancy

The next problem to analyze is whether a formula is redundant, for a fixed set of defaults. The complexity of formula redundancy w.r.t. faithful and consequence-equivalence is in Σ_3^p and Σ_4^p, respectively.

Theorem 23. *The problem of formula redundancy for faithful and consequence-equivalence is in Σ_3^p and Σ_4^p, respectively, for Reiter, justified, constrained, and rational default logic.*

Proof. Both problems can be expressed as the existence of a subset $W' \subset W$ such that W' is equivalent to W. Since verifying equivalence is in Π_2^p and Π_3^p, respectively, for faithful and consequence-equivalence, the claim follows. □

Regarding hardness, we first show a theorem characterizing the complexity of the problem for the case of faithful equivalence. We then show a more general technique allowing a hardness result to be raised one level in the polynomial hierarchy.

Theorem 24. *The problem of formula redundancy based on faithful equivalence is Σ_3^p-hard, and remains hard for consistent theories, for Reiter, justified, constrained, and rational default logic.*

Proof. We reduce the problem of validity of $\exists X \forall Y \exists Z.F$, where $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, and $Z = \{z_1, \ldots, z_n\}$ to the problem of redundancy of a formula. The default theory corresponding to the formula $\exists X \forall Y \exists Z.F$ is the theory $\langle D, W \rangle$ defined as follows, where a, $\{s_1, \ldots, s_n\}$, $\{r_1, \ldots, r_n\}$, $\{h_1, \ldots, h_n\}$, and $\{p_1, \ldots, p_n\}$ are new variables.

$$W = \{s_i \mid 1 \leqslant i \leqslant n\} \cup \{r_i \mid 1 \leqslant i \leqslant n\}$$

$$D = D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6$$

$$D_1 = \left\{ \frac{s_i \wedge r_i : \neg s_j}{a}, \frac{s_i \wedge r_i : \neg r_j}{a} \;\middle|\; \begin{array}{l} 1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant n \end{array} \right\}$$

$$D_2 = \left\{ \frac{: \neg s_i \wedge \neg r_i}{a} \;\middle|\; 1 \leqslant i \leqslant n \right\}$$

$$D_3 = \left\{ \frac{: y_i}{y_i \wedge h_i}, \frac{: \neg y_i}{\neg y_i \wedge h_i} \;\middle|\; 1 \leqslant i \leqslant n \right\}$$

$$D_4 = \left\{ \frac{: x_i}{p_i \wedge x_i}, \frac{: \neg x_i}{p_i \wedge \neg x_i} \;\middle|\; 1 \leqslant i \leqslant n \right\}$$

$$D_5 = \left\{ \frac{x_i \wedge r_i : \top}{\wedge W}, \frac{\neg x_i \wedge s_i : \top}{\wedge W} \;\middle|\; 1 \leqslant i \leqslant n \right\}$$

$$D_6 = \left\{ \frac{p_1 \wedge \cdots \wedge p_n \wedge h_1 \wedge \cdots \wedge h_n : F}{\wedge W} \right\}$$

The background theory W is consistent. The defaults of D_1 and D_2 cannot be applied from it. The defaults of D_3 and D_4 generate an extension for every possible truth evaluation over $X \cup Y$; this extension also contains all variables h_i and p_i. Whether or not the last default is applicable, its consequence is equivalent to the background theory.

Let $W' \subset W$. If there is an index i such that both s_i and r_i are in W', one of the defaults of D_1 is applicable, generating a. Therefore, W' is not equivalent to W. If there exists an index i such that neither s_i nor r_i is in W', the ith default of D_2 is applicable, still generating a.

In order to check for redundancy, we therefore only have to consider subsets $W' \subset W$ such that, for every $1 \leqslant i \leqslant n$, either $s_i \in W'$ or $r_i \in W'$ but not both. Let ω_X be the assignment on the variables X such that x_i is assigned to true if $s_i \in W'$ and to false if $r_i \in W'$. The defaults of D_3 and D_4 generate an arbitrary truth evaluation of the variables $X \cup Y$. If the assignment on X is not equal to ω_X, the formula $\wedge W$ is generated, thus leading to an extension that is also an extension of W. As a result, all extensions of W' that do not match the value ω_X are also extensions of W. If the same holds also for the extensions for which the values of X match ω_X, then W' is equivalent to W.

For a given W' we consider the extensions consistent with ω_X. There is exactly one such extension for each possible truth evaluation over Y. If the default of D_6 can be applied, it generates $\wedge W$, thus making W' equivalent to W. In turn, the default of D_6 can be applied for all truth evaluation over Y if and only if for all such truth evaluation, F is satisfiable. As a result, W' is equivalent to W if and only if, for all possible truth evaluations over Y, the formula F is satisfiable. Since there exists a relevant W' for each truth evaluation over X, the formula W is redundant if and only if there exists a truth evaluation over X such that, for all possible truth evaluations over Y, the formula F is satisfiable. □

In order to prove the Σ_4^p-hardness of the problem of formula redundancy under consequence-equivalence, we should provide a reduction from $\exists\forall\exists\forall$QBF validity into this problem. A simpler proof can however be given, based on the following consideration: checking clause redundancy has been proved Π_2^p-hard or Π_3^p-hard using reductions from QBFs that results in default theories having $W = \{a\}$ as the background theory. As a result, these reductions also prove that formula redundancy is Π_2^p-hard or Π_3^p-hard. In other words, we can reduce the validity of a $\forall\exists$QBF or a $\forall\exists\forall$QBF into the problems of formula redundancy. What we show is that, if such reductions satisfy some assumptions, we can obtain new reductions from QBFs having an additional existential quantifier in the front. The assumptions are that the default theory resulting from the reduction is such that:

1. the background theory that results from the reduction is classically irredundant;
2. the matrix of the QBF occurs only in the justification of a single default and does not affect the rest of the default theory.

The reductions used for proving the hardness of clause redundancy satisfy both assumptions. In particular, $\forall X \exists Y \forall Z.F$ is valid if and only if the background theory of the following theory is consequence-redundant, where D, α, β, γ, do not depend on F but only on the quantifiers of the QBF and W is classically irredundant.

$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F}{\gamma} \right\}, W \right\rangle$$

The fact that the matrix of the QBF is copied "verbatim" in the default theory is exploited as follows: if ω_w is a truth evaluation over the variable w, then $\forall X \exists Y \forall Z.F|_{\omega_w}$ is valid if and only if the background theory of $\langle D \cup \{\frac{\alpha:\beta \wedge F \wedge \omega_w}{\gamma}\}, W\rangle$ is redundant. This default theory can be modified in such a way that the subsets of the background theory are in correspondence with the truth evaluations over ω_w. This way, the resulting theory is redundant if and only if $\exists w \forall X \exists Y \forall Z\ .\ F$ is valid. The resulting default theory still satisfies the two assumptions above on the background theory and on the use of the matrix of the QBF; therefore, this procedure can be iterated to obtain a reduction from $\exists\forall\exists\forall$QBF validity into the problem of formula redundancy under consequence-equivalence. A similar technique can be used for faithful equivalence.

The details of this technique are in the following three lemmas. The first one shows that a literal can be moved from the justification of a default to the background theory and vice versa, under certain conditions.

Lemma 10. *Let W be a propositional formula and $D \cup \{d\}$ a set of defaults. Assume that l is a literal whose variable do not occur in W, D,* prec(d)*, and* cons(d) *(in other words, it may only occur in* just(d)*). Then, the selected processes of the first of following theory can be converted into a selected process of the second by replacing d with d', and vice versa. This holds for Reiter, justified, constrained, and rational default logic.*

$$\langle D \cup \{d\}, W \cup \{l\} \rangle$$
$$\langle D \cup \{d'\}, W \rangle$$

where

$$d' = \frac{\mathrm{prec}(d) : \mathrm{just}(d) \wedge l}{\mathrm{cons}(d)}$$

Proof. The variable in l only occurs in the background theory $W \cup \{l\}$ and in the justification of d and d'. The conditions for a process of the first theory being selected either involve $(W \cup \{l\}) \cup \mathrm{just}(d)$ or $W \cup \{l\}$ with other formulae not containing l, where W does not contain the variable of l. As a result, moving l from the background theory to the justification of d or vice versa does not affect these conditions.

Note that the processes are the same, but the extensions are different in that l is in all extensions of the first theory but in none of the second. □

The second lemma is an obvious consequence of the above: under the same conditions, moving a literal from the justification of a default to the background theory or vice versa does not change the redundancy of a theory.

Lemma 11. *Let W and W' be two propositional formulae such that $W' \subseteq W$, $D \cup \{d\}$ a set of defaults, and l a literal whose variable do not occur in W, D,* prec(d)*, and* cons(d) *(that is, it may only occur in* just(d)*). It holds $W' \equiv^e_{D'} W$ if and only if $W' \cup \{l\} \equiv^e_{D''} W \cup \{l\}$, where D' and D'' are as follows. This holds for Reiter, justified, constrained, and rational default logic.*

$$D' = D \cup \left\{ \frac{\mathrm{prec}(d) : \mathrm{just}(d) \wedge l}{\mathrm{cons}(d)} \right\}$$
$$D'' = D \cup \{d\}$$

Proof. Obvious consequence of the lemma above: $\langle D', W'\rangle$ and $\langle D', W\rangle$ have the same processes of $\langle D'', W' \cup \{l\}\rangle$ and $\langle D'', W \cup \{l\}\rangle$, respectively. $\Box$

A consequence of this lemma is that W is redundant according to D' if and only if $W \cup \{l\}$ is redundant according to D''. Indeed, l is not mentioned in the consequences of the defaults; therefore, a subset of $W \cup \{l\}$ can only be equivalent to $W \cup \{l\}$ if it contains l. The lemma is formulated in the more general way because it is necessary for proving the following lemma. The same property can be proved using consequence-equivalence because moving l from the justification of the default to the background theory has the only effect of adding l to all extensions.

Lemma 12. *If D is a set of defaults, w a variable, and W a classically irredundant formula, then there exists $W' \subset W$ such that $W' \cup \{w\} \equiv^e_D W \cup \{w\}$ or $W' \cup \{\neg w\} \equiv^e_D W \cup \{\neg w\}$ if and only if the following theory is redundant:*

$$\left\langle D_w \cup \left\{ \frac{p \wedge \alpha : \beta}{\gamma} \;\middle|\; \frac{\alpha : \beta}{\gamma} \in D \right\}, W \cup \{w^+, w^-\} \right\rangle$$

where:

$$D_w = \left\{ \frac{w^+ \wedge w^- : \neg W}{\neg p}, \frac{: \neg w^+ \wedge \neg w^-}{\neg p}, \frac{w^+ : w \wedge p}{w \wedge p}, \frac{w^- : \neg w \wedge p}{\neg w \wedge p} \right\}$$

and w^+, w^-, and p are new variables. This holds for Reiter, justified, constrained, and rational default logic.

Proof. Since w^+ and w^- are new variables not contained in W and W is classically irredundant, $W \cup \{w^+, w^-\}$ is classically irredundant as well.

We now consider the processes that can be generated from $W \cup \{w^+, w^-\}$ and from its subsets. From $W \cup \{w^+, w^-\}$ we can apply only one of the last two defaults of D_w, generating either $w \wedge p$ or $\neg w \wedge p$. From this point on, we have exactly the same processes of $\langle D, W \cup \{w\}\rangle$ and $\langle D, W \cup \{\neg w\}\rangle$, the generated extensions only differing because of the addition of p, w^+, and w^-.

The proper subsets of $W \cup \{w^+, w^-\}$ are $W' \cup \{w^+, w^-\}$ where $W' \subset W$, $W' \cup \{w^+\}$, $W' \cup \{w^-\}$, and W', where $W' \subseteq W$. The fourth subset W' is not equivalent to W because the second default of D_w allows the derivation of $\neg p$, which is not derivable from W. If $W' \subset W$, since W is (classically) irredundant, $W' \cup \{w^+, w^-\}$ allows for the application of the first default of D_w, deriving $\neg p$; therefore, this subset is not equivalent to the background theory.

The only two other subsets to consider are $W' \cup \{w^+\}$ and $W' \cup \{w^-\}$. In the first subset, only $w \wedge p$ can be generated. In the second subset, only $\neg w \wedge p$ can be generated. From this point on, we have exactly the same processes of $W' \cup \{w\}$ and $W' \cup \{\neg w\}$ according to D. The generated extensions are the same but for the addition of p. $\Box$

These three lemmas together prove that a reduction from QBF to formula redundancy can be "raised" by the addition of an existential quantifier in the front of the QBF. Formally, this is proved by assuming the existence of a reduction from a QBF to the problem of irredundancy of a default theory in a given form, and shown the existence of a reduction from a QBF with an additional existential quantifier in the front to the problem of redundancy of a default theory in the same form. We formally define this form as follows.

Definition 25 *(Verbatim reduction).* A reduction from the validity of a class of QBFs to the problem of formula redundancy of a default theory is a verbatim reduction if each formula $Q.E$, where Q is a sequence of quantifiers, is reduced to a default theory $\langle D, W\rangle$ where

$$D = D' \cup \left\{ \frac{\alpha : \beta \wedge E}{\gamma} \right\}$$

W is classically irredundant and W, D', α, β, and γ do not depend on E but only on Q.

This definition is instrumental to proving that a reduction from QBF to formula redundancy can be added an existential quantifier.

Lemma 13. *If there exists a verbatim polynomial reduction from QBF formulae whose sequence of quantifiers is in a given set S to the problem of formula redundancy of a default theory, then there exists a verbatim polynomial reduction* *from QBF formulae whose sequence of quantifiers is in the class $\{\exists T \mid T \in S\}$ to the problem of formula redundancy of a default theory. This holds for Reiter, justified, constrained, and rational default logic.*

Proof. In order to show the existence of the reduction, let $\exists w Q.F$ be an arbitrary QBF formula whose sequence of quantifiers is in $\{\exists T \mid T \in S\}$.

By assumption, both $Q.F|_{w=\mathsf{true}}$ and $Q.F|_{w=\mathsf{false}}$ can be reduced to the problem of formula redundancy by a verbatim reduction. These two QBF formulae only differ on their matrices, which are $F|_{w=\mathsf{true}}$ and $F|_{w=\mathsf{false}}$. Therefore, the resulting default theories are:

$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F|_{w=\mathsf{true}}}{\gamma} \right\}, W \right\rangle$$
$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F|_{w=\mathsf{false}}}{\gamma} \right\}, W \right\rangle$$

Since w does not occur anywhere in these defaults theories, we can replace $F|_{w=\mathsf{true}}$ and $F|_{w=\mathsf{false}}$ with $F \wedge w$ and $F \wedge \neg w$, respectively. Indeed, justifications are only checked for consistency, and for any formula R not containing w, the consistency of $R \cup F|_{w=\mathsf{true}}$ is the same as the consistency of $R \cup (F \wedge w)$, and the consistency of $R \cup F|_{w=\mathsf{false}}$ is the same as the consistency of $R \cup (F \wedge \neg w)$.

$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F \wedge w}{\gamma} \right\}, W \right\rangle$$
$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F \wedge \neg w}{\gamma} \right\}, W \right\rangle$$

By Lemma 11, formula redundancy of these two theories corresponds to formula redundancy of the same theories with w or $\neg w$ moved to the background theory. More precisely, the redundancy of the first theory correspond to the existence of a subset $W' \subset W$ such that $W' \cup \{w\}$ is equivalent to $W \cup \{w\}$ according to the defaults $D \cup \{\frac{\alpha:\beta\wedge F}{\gamma}\}$. The same holds for the second theory.

$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F}{\gamma} \right\}, W \cup \{w\} \right\rangle$$
$$\left\langle D \cup \left\{ \frac{\alpha : \beta \wedge F}{\gamma} \right\}, W \cup \{\neg w\} \right\rangle$$

By Lemma 12, since W is classically irredundant, we have that either the first or the second of the two theories are redundant if and only if the following theory is redundant:

$$\left\langle D_w \cup \left\{ \frac{p \wedge \alpha' : \beta'}{\gamma'} \;\middle|\; \frac{\alpha' : \beta'}{\gamma'} \in D \cup \left\{ \frac{\alpha : \beta \wedge F}{\gamma} \right\} \right\}, W \cup \{w^+, w^-\} \right\rangle$$

where D_w is defined in the statement of Lemma 12. As a result, this formula is redundant if and only if either $Q.F|_{w=\mathsf{true}}$ is valid or $Q.F|_{w=\mathsf{false}}$ is valid, that is, $\exists w Q.F$ is valid.

In order to complete the lemma, we have to show that this reduction from an arbitrary QBF $\exists w Q.F$ to the above default theory is a verbatim reduction. Since W is classically irredundant by assumption and w^+ and w^- are new variables, $W \cup \{w^+, w^-\}$ is classically irredundant as well. In the above theory, the matrix F of the QBF only affects the justification of the default $\frac{p\wedge\alpha:\beta\wedge F}{\gamma}$. Therefore, the reduction is a verbatim reduction. □

The above lemmas are also valid for consequence-equivalence. In both cases, we have that the hardness of formula redundancy is one level higher in the polynomial hierarchy than clause redundancy.

Theorem 25. *Formula redundancy is Σ_3^p-hard for faithful equivalence and Σ_4^p-hard for consequence-equivalence in Reiter, justified, constrained, and rational default logic.*

Proof. The reduction shown after Theorem 19 and the reduction used in Theorem 22 are reductions from $\forall\exists$QBF and $\forall\exists\forall$QBF, respectively, into the problem of formula redundancy. These reductions produce a default theory in which the background theory contains a single non-tautological clause, and is therefore classically irredundant, and the matrix of the QBF only occurs in the justification of a single default. These are the conditions of Lemma 13. As a result, one can reduce an ∃∀∃QBF or an ∃∀∃∀QBF to the problem of formula redundancy by iteratively applying the modification of Lemma 13 for all variables of the first existential quantifier. These reductions are polynomial because the repeated application of Lemma 13 does not increase the size of the involved default theories more than polynomially: every application adds two literals to background theory, four new defaults (each of size linear in the size of the background theory), and a new symbol in each default. □

D.3. Redundancy of defaults

The following lemma is the version of Theorem 11 in the case of default redundancy rather than clause redundancy. It proves that some defaults can be made irredundant while not changing the redundancy status of the others. Such results are generally useful to build default theories having some specific properties w.r.t. redundancy. In particular, the following lemma will be later used in Theorem 14 to prove that the problems of redundancy of clauses can be reduced to the corresponding problems of redundancy of defaults.

Lemma 5 *(proof). For every default theory $\langle D, W\rangle$ and every set of defaults $D_I \subseteq D$, let p and q two new variables and $\{v_i\}$ be a set of new variables in bijective correspondence with the defaults of D_I, and let D_1, D_2, D_3 be defined as follows*:

$$D_1 = \{d+, d-\}$$

where:

$$d+ = \frac{: p \wedge q}{p \wedge q}$$

$$d- = \frac{: \neg p \wedge q}{\neg p \wedge q}$$

$$D_2 = \left\{ \frac{q \wedge (\neg p \vee \alpha) : \neg p \vee \beta}{(p \vee v_i) \wedge (\neg p \vee \gamma)} \;\middle|\; \frac{\alpha : \beta}{\gamma} \in D_I,\, v_i \text{ is the variable corresponding to } \frac{\alpha : \beta}{\gamma} \right\}$$

$$D_3 = \left\{ \frac{q \wedge p \wedge \alpha : \beta}{\gamma} \;\middle|\; \frac{\alpha : \beta}{\gamma} \in D \backslash D_I \right\}$$

if $\langle D, W\rangle$ has extensions and W is consistent, in Reiter, justified, constrained, and rational default logics it holds that:

1. *the processes of $\langle D_1 \cup D_2 \cup D_3, W\rangle$ are (modulo the transformation of the defaults) the same as of $\langle D, W\rangle$ with $d+$ added to the front and a number of processes composed of $d-$ and a sequence containing all defaults of D_2*;
2. *the extensions of $\langle D_1 \cup D_2 \cup D_3, W\rangle$ are the same as of $\langle D, W\rangle$ with $\{p, q\}$ added plus the single extension $\{\neg p, q\} \cup \{v_i\}$*;
3. *a subset of $D_1 \cup D_2 \cup D_3$ is faithfully equivalent to it if and only if it contains $D_1 \cup D_2$ and the set of original defaults corresponding to those of $D_2 \cup D_3$ is faithfully equivalent to D.*

Proof. Since all defaults of $D_2 \cup D_3$ have q as a precondition, they are not applicable from W. The only defaults that are applicable to W are therefore $d+$ and $d-$, which are mutually exclusive.

Let us consider the processes with $d-$ in first position. Since $d-$ generates $\neg p$, the defaults of D_3 are not applicable. We prove that $[d-] \cdot \Pi_2$ is a successful process, where Π_2 is an arbitrary sequence containing all defaults of D_2. The preconditions of all defaults of D_2 are entailed by $q \wedge \neg p$. The union of the justifications and consequences of all defaults of this process is $\{\neg p, q\} \cup \{\neg p \vee \beta, p \vee v_i, \neg p \vee \gamma\}$, which is equivalent to $\{\neg p, q\} \cup \{v_i\}$. This set is consistent with the background theory, which does not contain the variables p, q, and v_i.

If a subset of $D_1 \cup D_2 \cup D_3$ does not contain $d-$, the literal $\neg q$ cannot derived because no other default has $\neg q$ as a conclusion. If a subset of $D_1 \cup D_2 \cup D_3$ does not contain a default of D_2, the corresponding variable v_i is not in this extension. As a result, every subset of $D_1 \cup D_2 \cup D_3$ that is equivalent to it contains $\{d-\} \cup D_2$.

Let us now consider the processes with $d+$ in first position. Such a process cannot contain $d-$. Since p and q are generated, the defaults of $D_2 \cup D_3$ can be simplified to $\frac{\alpha : \beta}{\gamma}$ by removing all clauses containing p or q and all literals $\neg p$ and $\neg q$ from the clauses containing them. As a result, the processes having $d+$ in first position correspond to the processes of the original theory.

Provided that the original theory has extensions, every subset of $D_1 \cup D_2 \cup D_3$ not containing $d+$ lacks these extensions. The defaults of D_3 are redundant if and only if they are redundant in the original theory. More precisely, a subset $D' \subset D_1 \cup D_2 \cup D_3$ is equivalent to $D_1 \cup D_2 \cup D_3$ if and only if D' contains $D_1 \cup D_2$, and the set of original defaults D'' corresponding to the defaults of $D' \cap (D_2 \cup D_3)$ is equivalent to D. □

The following counterexample shows that Reiter and rational default logic do not have the unitary redundancy property w.r.t. redundancy of defaults.

Counterexample 7 *(proof). There exists a set of defaults D such that, according to Reiter and rational default logic*:

1. *for every $d \in D$, the theory $\langle D \backslash \{d\}, \emptyset\rangle$ has extensions and $D \backslash \{d\} \not\equiv^e_\emptyset D$;*
2. *there exists $D' \subset D$ such that $D' \equiv^e_\emptyset D$.*

Proof. We use a pair of defaults that lead to failure is they are together in the same process. Removing one of them from the default theory leads to a new extension, while removing both of them lead to the original set of extensions. The following defaults are a realization of this idea.

$$D = \{d_1, d_2, d_3, d_4\}$$

where:

$$d_1 = \frac{: b}{b \wedge c}$$
$$d_2 = \frac{: b}{b \wedge \neg c}$$
$$d_3 = \frac{: \neg b \wedge c}{\neg b \wedge c}$$
$$d_4 = \frac{: \neg b \wedge \neg c}{\neg b \wedge \neg c}$$

The extensions of some $\langle D', \emptyset\rangle$, with $D' \subseteq D$, are as follows:

$D' = D$ we can either apply both d_1 and d_2 (leading to a failure) or either d_3 or d_4, but not both; the extensions of this theory therefore are $Cn(\{\neg b, c\})$ and $Cn(\{\neg b, \neg c\})$;

$D' = \{d_1, d_3, d_4\}$ one among d_1, d_3, and d_4 can be applied, but not any two of them at the same time; that results in three processes, generating the extensions $Cn(\{b, c\})$, $Cn(\{\neg b, c\})$, and $Cn(\{\neg b, \neg c\})$;

$D' = \{d_2, d_3, d_4\}$ similar to the previous case: the generated extensions are $Cn(\{b, \neg c\})$, $Cn(\{\neg b, c\})$, and $Cn(\{\neg b, \neg c\})$;

$D' = \{d_1, d_2, d_3\}$ either d_1, d_2 or d_3 can be applied; the first two lead to failure; the only extension of this theory is therefore $Cn(\{\neg b, c\})$;

$D' = \{d_1, d_2, d_4\}$ similar to the case above: the only extension is $Cn(\{\neg b, \neg c\})$;

$D' = \{d_3, d_4\}$ both d_3 and d_4 can be applied, but not at the same time; as a result, the generated extensions are $Cn(\{\neg b, c\})$ and $Cn(\{\neg b, \neg c\})$.

As a result, $\langle D \backslash \{d\}, \emptyset\rangle$ has extensions for every $d \in D$, but its set of extensions is different from that of $\langle D, \emptyset\rangle$. On the other hand, $\langle D, \emptyset\rangle$ and $\langle \{d_3, d_4\}, \emptyset\rangle$ have the same extensions. □

The same result holds for constrained default logic.

Counterexample 8 *(proof). There exists a set of defaults D such that, according to constrained default logic*:

1. *for every $d \in D$, it holds $D \backslash \{d\} \not\equiv^e_\emptyset D$;*
2. *there exists $D' \subset D$ such that $D' \equiv^e_\emptyset D$.*

Proof. The defaults are the following ones:

$$D = \{d_1, d_2, d_3\}$$

where:

$$d_1 = \frac{:x}{a}$$
$$d_2 = \frac{:x}{b}$$
$$d_3 = \frac{:\neg x \wedge \neg y}{a \wedge b}$$

The theory $\langle D, \emptyset\rangle$ has two selected processes (modulo permutation of defaults): $[d_1, d_2]$ and $[d_3]$, both generating the extension $Cn(\{a, b\})$. Removing either d_1 or d_2 causes the first process to become $[d_1]$ or $[d_2]$, thus creating a new extension $Cn(\{a\})$ or $Cn(\{b\})$, respectively. On the other hand, removing both d_1 and d_2 makes the only remaining process to be $[d_3]$, which generates the same single extension $Cn(\{a, b\})$ of the original theory. The default d_3 is not redundant, but can be made so by applying the transformation of Lemma 5. □

Justified default logic has the unitary redundancy property w.r.t. default redundancy. The proof requires two lemmas. The first one is about extendibility of processes when new defaults are added to a theory.

Lemma 14. *In justified default logic, if Π is a selected process of a default theory $\langle D', W\rangle$ and $D' \subseteq D$, then there exists a sequence Π' of defaults of $D \backslash D'$ such that $\Pi \cdot \Pi'$ is a selected process of $\langle D, W\rangle$.*

Proof. Let Π be a selected process of $\langle D', W\rangle$. By definition, it holds $W \cup \mathrm{cons}(\Pi[d]) \models \mathrm{prec}(d)$ and $W \cup \mathrm{cons}(\Pi) \top \mathrm{just}(d)$ for every $d \in \Pi$. As a result, Π is a also a successful process of $\langle D, W\rangle$. Therefore, there exists Π' such that $\Pi \cdot \Pi'$ is a selected process of $\langle D, W\rangle$ because justified default logic is failsafe [29]. If Π' contains defaults of D', then Π would not be a closed process of $\langle D', W\rangle$. □

In order for proving the second lemma, we need an intermediate result.

Lemma 15. *In justified default logic, the selected processes of a default theory $\langle D, W\rangle$ generating the extension E are composed of exactly the defaults of the following set*:

$$GEN(E, D) = \{d \in D \mid E \models \mathrm{prec}(d) \text{ and } E \top \mathrm{just}(d) \cup \mathrm{cons}(d)\}$$

Proof. Assume that Π is a selected process generating E that does not contain a default $d \in GEN(E, D)$. Since $E \models \mathrm{prec}(d)$, $E \top \mathrm{just}(d) \cup \mathrm{cons}(d)$, and $E = W \cup \mathrm{cons}(\Pi)$, we have that $W \cup \mathrm{cons}(\Pi) \models \mathrm{prec}(d)$ and $W \cup \mathrm{cons}(\Pi) \top \mathrm{just}(d) \cup \mathrm{cons}(d)$. As a result, $\Pi \cdot [d]$ is a successful process, contradicting the assumption.

Let Π be a selected process containing a default d not in $GEN(E, D)$. By definition, either $E \not\models \mathrm{prec}(d)$ or $E \perp \mathrm{just}(d) \cup \mathrm{cons}(d)$. The first condition implies that $W \cup \mathrm{cons}(\Pi[d]) \not\models d$ whichever the position of d in Π is. The second condition implies $W \cup \mathrm{cons}(\Pi) \perp \mathrm{just}(d) \cup \mathrm{cons}(d)$: the process Π is not successful contrary to the assumption. □

The next lemma relates the processes of two theories when they are assumed to have the same extension. In this lemma and in the following theorem, when a process is used in a place where a set of defaults is expected, it means the set of defaults of the process. For example, if Π is a sequence of defaults and D' a set of defaults, $\Pi \cap D'$ is the set of defaults that are both in Π and in D.

Lemma 16. *In justified default logic, if W is a formula and D and D' are two sets of default such that $D' \subseteq D$, $D' \equiv^e_W D$, and Π is a selected process of $\langle D, W\rangle$, then there exists a selected process of $\langle D', W\rangle$ made exactly of the defaults of $\Pi \cap D'$ and generating the same extension as generated by Π.*

Proof. Let $E = W \cup \text{cons}(\Pi)$ be the extension that is generated by Π. By the lemma above, it is generated by the defaults in $GEN(E, D)$. Since E is also an extension of $\langle D', W\rangle$, it is generated by a process Π' made exactly of the defaults of $GEN(E, D') = GEN(E, D) \cap D' = \Pi \cap D'$. □

The above lemmas allows to prove that justified default logic has the monotonic redundancy property w.r.t. sets of defaults and faithful equivalence.

Theorem 13 *(proof). Faithful equivalence of sets of defaults for justified default logic has the monotonic redundancy property.*

Proof. We prove that, if $D' \subseteq D'' \subseteq D$ and $D' \equiv^e_W D$ then $D'' \equiv^e_W D$ for justified default logic. We first show that every extension of $\langle D'', W\rangle$ is also an extension of $\langle D, W\rangle$, and then show the converse.

Let E be an extension of $\langle D'', W\rangle$. Let Π be one its generating processes. By Lemma 14, there exists a sequence Π' of defaults of $D\backslash D''$ such that $\Pi \cdot \Pi'$ is a selected process of $\langle D, W\rangle$. Let E' be its generated extension. Since E is generated by Π and E' is generated by $\Pi \cdot \Pi'$, we have $E' \models E$. We prove that $E \models E'$, which implies $E \equiv E'$.

By Lemma 16, since Π is a selected process of $\langle D, W\rangle$ and this theory is faithfully equivalent to $\langle D', W\rangle$, there exists a selected process Π'' of $\langle D', W\rangle$ made of the defaults of $(\Pi \cdot \Pi') \cap D'$ and generating the extension E'. Since Π' is made of defaults of $D\backslash D''$ and $D' \subseteq D''$, we have that $(\Pi \cdot \Pi') \cap D' = \Pi \cap D'$. As a result, Π'' is only made of defaults in $\Pi \cap D'$. Since Π'' generates E' and Π generates E, we have $E \models E'$. We can therefore conclude that $E \equiv E'$.

Let us now prove the converse: we assume that E is an extension of $\langle D, W\rangle$ and prove that it is also an extension of $\langle D'', W\rangle$. Let Π be the process of $\langle D, W\rangle$ that generates E. By definition, the following two properties are true:

1. $W \cup \text{cons}(\Pi) \models \text{prec}(d)$ for every $d \in \Pi$;
2. $W \cup \text{cons}(\Pi) \bot \text{just}(d) \cup \text{cons}(d)$ for every $d \notin \Pi$.

By Lemma 16, the theory $\langle D', W\rangle$ has a selected process Π' that is composed exactly of the defaults of $\Pi \cap D'$ and that generates the same extension E. Since $W \cup \text{cons}(\Pi') \equiv W \cup \text{cons}(\Pi)$, the two properties are equivalent to the following two ones:

1. $W \cup \text{cons}(\Pi') \models \text{prec}(d)$ for every $d \in \Pi$;
2. $W \cup \text{cons}(\Pi') \bot \text{just}(d) \cup \text{cons}(d)$ for every $d \notin \Pi$.

The first property implies that every default $d \in \Pi \cap (D''\backslash D')$ is applicable to Π': this is because the precondition of d is entailed by $W \cup \text{cons}(\Pi')$ and the process $\Pi' \cdot [d]$ is successful because so is Π, which contains all default of $\Pi' \cdot [d]$. The second property implies that no default of $D''\backslash\Pi$ is applicable to Π'. As a result, Π' and the sequence composed of all defaults of $\Pi \cap (D''\backslash D')$ in any order form a selected process of D''. The extension generated by this process is equivalent to E because this process is composed of a superset of the defaults of Π' and a subset of the defaults of Π, and these two processes both generate E. □

References

[1] G. Ausiello, A. D'Atri, D. Saccà, Minimal representation of directed hypergraphs, SIAM Journal on Computing 15 (2) (1986) 418–431.

[2] G. Antoniou, A tutorial on default logics, ACM Computing Surveys 31 (4) (1999) 337–359.

[3] G. Antoniou, V. Sperschneider, Operational concepts of non-monotonic logics, part 1: Default logic, Artificial Intelligence Review 8 (1) (1994) 3–16.

[4] P. Besnard, An Introduction to Default Logic, Springer, Berlin, 1989.

[5] R. Bruni, Approximating minimal unsatisfiable subformulae by means of adaptive core search, Discrete Applied Mathematics 130 (2) (2003) 85–100.

[6] P. Bossu, P. Siegel, Saturation, non-monotonic reasoning and the closed world assumption, Artificial Intelligence 25 (1985) 16–63.

[7] H. Büning, X. Zhao, Extension and equivalence problems for clause minimal formulae, Annals of Mathematics and Artificial Intelligence 43 (1) (2005) 295–306.

[8] M. Cadoli, T. Eiter, G. Gottlob, Default logic as a query language, IEEE Transactions on Knowledge and Data Engineering 9 (3) (1997) 448–463.
[9] J. de Kleer, K. Konolige, Eliminating the fixed predicates from a circumscription, Artificial Intelligence 39 (1989) 391–398.
[10] J. Delgrande, T. Schaub, Expressing default logic variants in default logic, Journal of Logic and Computation 15 (5) (2005) 593–621.
[11] J.P. Delgrande, T. Schaub, W.K. Jackson, Alternative approaches to default logic, Artificial Intelligence 70 (1994) 167–237.
[12] T. Eiter, M. Fink, H. Tompits, S. Woltran, Strong and uniform equivalence in answer-set programming: Characterizations and complexity results for the non-ground case, in: Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005), 2005, pp. 695–700.
[13] T. Eiter, G. Gottlob, Propositional circumscription and extended closed world reasoning are Π_2^p-complete, Theoretical Computer Science 114 (1993) 231–245.
[14] H. Fleischner, O. Kullmann, S. Szeider, Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference, Theoretical Computer Science 289 (2002) 503–516.
[15] C. Froidevaux, J. Mengin, A framework for default logics, in: European Workshop on Logics in AI (JELIA'92), 1992, pp. 154–173.
[16] C. Froidevaux, J. Mengin, Default logics: A unified view, Computational Intelligence 10 (1994) 331–369.
[17] G. Gottlob, C.G. Fermüller, Removing redundancy from a clause, Artificial Intelligence 61 (1993) 263–289.
[18] A. Ginsberg, Knowledge base reduction: A new approach to checking knowledge bases for inconsistency & redundancy, in: Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88), 1988, pp. 585–589.
[19] L. Giordano, A. Martelli, On cumulative default logics, Artificial Intelligence 66 (1994) 161–179.
[20] G. Gottlob, Complexity results for non-monotonic logics, Journal of Logic and Computation 2 (1992) 397–425.
[21] G. Gottlob, Translating default logic into standard autoepistemic logic, Journal of the ACM 42 (1995) 711–740.
[22] P. Hammer, A. Kogan, Optimal compression of propositional Horn knowledge bases: Complexity and approximation, Artificial Intelligence 64 (1) (1993) 131–145.
[23] E. Hemaspaandra, G. Wechsung, The minimization problem for Boolean formulas, in: Proceedings of the Thirty Eighth Annual Symposium on the Foundations of Computer Science (FOCS'97), 1997, pp. 575–584.
[24] T. Janhunen, Evaluating the effect of semi-normality on the expressiveness of defaults, Artificial Intelligence 144 (2003) 233–250.
[25] L. Kirousis, P. Kolaitis, A dichotomy in the complexity of propositional circumscription, in: Proceedings of the Nineteenth IEEE Symposium on Logic in Computer Science (LICS 2004), 2001, pp. 71–80.
[26] P. Liberatore, Redundancy in logic I: CNF propositional formulae, Artificial Intelligence 163 (2) (2005) 203–232.
[27] P. Liberatore, Representability in default logic, Logic Journal of the IGPL 13 (3) (2005) 335–351.
[28] P. Liberatore, On the complexity of extension checking in default logic, Information Processing Letters 98 (2) (2006) 61–65.
[29] P. Liberatore, Where fail-safe default logics fail, ACM Transactions on Computational Logic 8 (2) (2007).
[30] P. Liberatore, Redundancy in logic II: 2-CNF and Horn propositional formulae, Artificial Intelligence 172 (2–3) (2008) 265–299.
[31] J. Lang, P. Marquis, In search of the right extension, in: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), 2000, pp. 625–636.
[32] V. Lifschitz, D. Pearce, A. Valverde, Strongly equivalent logic programs, ACM Transactions on Computational Logic 2 (4) (2001) 526–541.
[33] W. Lukaszewicz, Considerations on default logic: An alternative approach, Computational Intelligence 4 (1) (1988) 1–16.
[34] D. Maier, Minimum covers in relational database model, Journal of the ACM 27 (4) (1980) 664–674.
[35] D. Makinson, General theory of cumulative inference, in: Proceedings of the Second International Workshop on Non-Monotonic Reasoning, Springer, 1988, pp. 1–18.
[36] J. McCarthy, Circumscription—A form of non-monotonic reasoning, Artificial Intelligence 13 (1980) 27–39.
[37] A. Meyer, L. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, in: Proceedings of the Thirteenth Annual Symposium on Switching and Automata Theory (FOCS'72), 1972, pp. 125–129.
[38] W. Marek, M. Truszczyński, Non-Monotonic Logics: Context-Dependent Reasoning, Springer, Berlin, 1993.
[39] A. Mikitiuk, M. Truszczynski, Constrained and rational default logics, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95), 1995, pp. 1509–1517.
[40] G. Nordh, P. Jonsson, An algebraic approach to the complexity of propositional circumscription, in: Proceedings of the Nineteenth IEEE Symposium on Logic in Computer Science (LICS 2004), 2004, pp. 367–376.
[41] D.L. Poole, A logical framework for default reasoning, Artificial Intelligence 36 (1988) 27–47.
[42] D. Poole, Default logic, in: Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Non-Monotonic and Uncertainty Reasoning, Oxford University Press, Oxford, 1994, pp. 189–215.
[43] C. Papadimitriou, D. Wolfe, The complexity of facets resolved, Journal of Computer and System Sciences 37 (1988) 2–13.
[44] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1980) 81–132.
[45] R. Rosati, Model checking for non-monotonic logics, in: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99), 1999, pp. 76–83.
[46] T. Schaub, On constrained default theories, in: Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92), 1992, pp. 304–308.
[47] P. Siegel, L. Forget, A representation theorem for preferential logics, in: Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR'96), 1996, pp. 453–460.
[48] J. Schmolze, W. Snyder, Detecting redundant production rules, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI'97), 1997, pp. 417–423.
[49] J. Stillman, The complexity of propositional default logics, in: Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92), 1992, pp. 794–799.

[50] M. Truszczynski, Strong and uniform equivalence of non-monotonic theories—an algebraic approach, Annals of Mathematics and Artificial Intelligence 48 (3–4) (2006) 245–265.
[51] H. Turner, Strong equivalence for logic programs and default theories (made easy), in: Proceedings of the Sixth International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'01), 2001, pp. 81–92.
[52] C. Umans, The minimum equivalent DNF problem and shortest implicants, in: Proceedings of the Thirty Ninth Annual Symposium on the Foundations of Computer Science (FOCS'98), 1998, pp. 556–563.