TLP **13** (4–5): Online Supplement, July 2013. © 2013 [MICHAEL J. MAHER]
URL: http://dx.doi.org/10.1017/S1471068413000367

Relative expressiveness of defeasible logics II

MICHAEL J. MAHER

School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia
(*e-mail:* m.maher@adfa.edu.au)

submitted 10 April 2013; accepted 23 May 2013

Abstract

Maher (2012) introduced an approach for relative expressiveness of defeasible logics, and two notions of relative expressiveness were investigated. Using the first of these definitions of relative expressiveness, we show that all the defeasible logics in the **DL** framework are equally expressive under this formulation of relative expressiveness. The second formulation of relative expressiveness is stronger than the first. However, we show that logics incorporating individual defeat are equally expressive as the corresponding logics with team defeat. Thus the only differences in expressiveness of logics in **DL** arise from differences in how ambiguity is handled. This completes the study of relative expressiveness in **DL** begun in Maher (2012).

KEYWORDS: defeasible logic, non-monotonic reasoning, relative expressiveness

1 Introduction

Defeasible logics provide several linguistic features to support the expression of defeasible knowledge. There are also a variety of such logics, supporting different intuitions about reasoning in a defeasible setting. The **DL** framework (Antoniou *et al.* 2000; Billington *et al.* 2010) provides logics that allow ambiguity in the "truth" status of a literal to propagate, and logics that block ambiguity; it has logics that require an individual rule to defeat all competitors, and logics that allow a "team" of rules to defeat competitors. Given the different inferences supported by the different logics, it is interesting to determine whether these logics are equally powerful or whether, perhaps, some are more powerful than the others.

In terms of inference strength, (Billington *et al.* 2010) established the relationship between the different logics of **DL**. In terms of computational complexity, the logics of **DL** are equivalent: all have linear complexity (Maher 2001; Billington *et al.* 2010). Relative expressiveness of the different logics was first investigated in Maher (2012), which developed a framework, based on simulation in the presence of additional elements. Two notions of relative expressiveness within this framework were investigated: polynomial simulation wrt the addition of facts, and simulation wrt the addition of rules.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

In this paper we continue this investigation. We will see that all the logics of **DL** are equally expressive, using the first notion of relative expressiveness. Thus we cannot distinguish the logics based on this notion. We also establish that individual defeat has equal expressiveness to team defeat in the logics of **DL** wrt addition of rules. This is somewhat surprising, given the apparent greater sophistication of the team defeat inference rules. Given results in Maher (2012), this completes the study of relative expressiveness for **DL**.

The next two sections summarize the **DL** framework of defeasible logics and the notions of relative expressiveness introduced in Maher (2012). Then the following two sections together provide the proof that the logics of **DL** are of equal expressivity (in terms of simulation wrt addition of facts). The first shows the simulation of an ambiguity propagating logic by an ambiguity blocking logic, while the second shows a simulation in the reverse direction. Combined with results of Maher (2012), this establishes that the logics in **DL** all have the same expressiveness in this formulation.

The following sections investigate relative expressiveness via the second, stronger formulation. Adapting simulations of Maher (2012) to the stronger formulation, we establish that individual defeat has equal expressiveness to team defeat in the logics of **DL**. Proofs of the results in this paper are detailed and lengthy. They appear in an appendix.

2 Defeasible logic

In this section we can only present an outline of the defeasible logics we investigate. Further details can be obtained from (Billington *et al.* 2010) and the references therein. We address propositional defeasible logics, but the results should extend to a first-order language.

A defeasible theory is built from a language Σ of literals (which we assume is closed under negation) and a language Λ of labels. A *defeasible theory* $D = (F, R, >)$ consists of a set of facts F, a finite set of rules R, each rule with a distinct label from Λ, and an acyclic relation $>$ on Λ called the *superiority relation*. This syntax is uniform for all the logics considered here. Facts are individual literals expressing indisputable truths. Rules relate a set of literals (the body), via an arrow, to a literal (the head), and are one of three types: a strict rule, with arrow $\rightarrow$; a defeasible rule, with arrow $\Rightarrow$; or a defeater, with arrow $\rightsquigarrow$. Strict rules represent inferences that are unequivocally sound if based on definite knowledge; defeasible rules represent inferences that are generally sound. Inferences suggested by a defeasible rule may fail, due to the presence in the theory of other rules. Defeaters do not support inferences, but may impede inferences suggested by other rules. The superiority relation provides a local priority on rules. Strict or defeasible rules whose bodies are established defeasibly represent claims for the head of the rule to be concluded. The superiority relation contributes to the adjudication of these claims by an inference rule, leading (possibly) to a conclusion. Given a theory D, the corresponding languages are expressed by $\Sigma(D)$ and $\Lambda(D)$.

Defeasible logics derive conclusions that are outside the syntax of the theories. Conclusions may have the form $+dq$, which denotes that under the inference rule d

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

the literal q can be concluded, or $-dq$, which denotes that the logic can establish that under the inference rule d the literal q cannot be concluded. The syntactic element d is called a tag. In general, neither conclusion may be derivable: q cannot be concluded under d, but the logic is unable to establish that. Tags $+\Delta$ and $-\Delta$ represent monotonic provability (and unprovability) where inference is based on facts, strict rules, and modus ponens. We assume these tags and their inference rules are present in every defeasible logic. What distinguishes a logic is the inference rule for defeasible reasoning. The four logics discussed in the Introduction correspond to four different pairs of inference rules, labelled ∂, δ, ∂^*, and δ^*; they produce conclusions of the form (respectively) $+\partial q$, $-\partial q$, $+\delta q$, $-\delta q$, etc. The inference rules δ and δ^* require auxiliary tags and inference rules, denoted by σ and σ^*, respectively. For each of the four principal defeasible tags d, the corresponding logic is denoted by **DL**(d).

The four principal tags and corresponding inference rules represent different intuitions about defeasible reasoning: in ∂ and ∂^* ambiguity is blocked, while in δ and δ^* ambiguity is propagated; in ∂ and δ rules for a literal act as a team to overcome competing rules, while in ∂^* and δ^* a single rule must overcome all competing rules. A more detailed discussion of ambiguity and team defeat in the **DL** framework is given in Billington *et al.* (2010) and Maher (2012).

The inference rules are presented in the appendix in the form of the definition of a function $\mathscr{T}_D$ for a given theory D. Given a defeasible theory D, for any set of conclusions E, $\mathscr{T}_D(E)$ denotes the set of conclusions inferred from E using D and one application of an inference rule. $\mathscr{T}_D$ is a monotonic function on the complete lattice of sets of conclusions ordered by containment. The least fixedpoint of $\mathscr{T}_D$ is the set of all conclusions that can be drawn from D. We follow standard notation in that $\mathscr{T}_D \uparrow 0 = \emptyset$ and $\mathscr{T}_D \uparrow (n+1) = \mathscr{T}_D(\mathscr{T}_D \uparrow n)$.

The relative inference strength of the different logics in **DL** was established in the inclusion theorem of Billington *et al.* (2010). For any tag d, $+d(D)$ denotes the set of conclusions of D of the form $+dq$ and similarly for $-d$.

Theorem 1 (*Inclusion Theorem (Billington et al. 2010)*)
Let D be a defeasible theory.

(a) $+\Delta(D) \subseteq +\delta^*(D) \subseteq +\delta(D) \subseteq +\partial(D) \subseteq +\sigma(D) \subseteq +\sigma^*(D)$.
(b) $-\sigma^*(D) \subseteq -\sigma(D) \subseteq -\partial(D) \subseteq -\delta(D) \subseteq -\delta^*(D) \subseteq -\Delta(D)$.
(c) $+\delta^*(D) \subseteq +\partial^*(D) \subseteq +\sigma^*(D)$
(d) $-\sigma^*(D) \subseteq -\partial^*(D) \subseteq -\delta^*(D)$

Parts (a) and (b) are proved in Billington *et al.* (2010). Parts (c) and (d) can be established by similar methods.

3 Simulating defeasible logics

Maher (2012) introduced a framework for addressing the relative expressiveness of defeasible logics. The framework identifies the greater (or equal) expressiveness of L_2 compared to L_1 with the ability to simulate any theory D in a logic L_1 by a

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

theory $T(D)$ in the logic L_2. Simple simulation was shown not to be sufficiently discriminating, so simulation was required to hold in the presence of an addition to the theory.

The *addition* of a theory A to a theory D is denoted by $D + A$. Addition is essentially the union of the theories, but we require $\Lambda(D) \cap \Lambda(A) = \emptyset$, so that the addition of theories preserves the property that distinct rules have distinct labels. This requirement also has the effect that a superiority statement in D cannot affect a rule in A, and vice versa. Let $D = (F, R, >)$ and $A = (F', R', >')$. Then $D+A = (F \cup F', R \cup R', > \cup >')$. $\Lambda(D+A) = \Lambda(D) \cup \Lambda(A)$ and $\Sigma(D+A) = \Sigma(D) \cup \Sigma(A)$.

A simulating theory $T(D)$ in general will involve additional literals, rules and labels beyond those of D. If additions A were permitted to affect these, the notion of simulation would become trivial, so we restrict additions to have only an indirect effect on $T(D)$, via $\Sigma(D)$. Given a theory D and a possible simulating theory $T(D)$, we say an addition A is *modular* if $\Sigma(A) \cap \Sigma(T(D)) \subseteq \Sigma(D)$, $\Lambda(D) \cap \Lambda(A) = \emptyset$, and $\Lambda(T(D)) \cap \Lambda(A) = \emptyset$. In general, we will consider a class of additions but for any D and $T(D)$ only the modular additions in the class will be considered.

Since different logics involve different tags, conclusions from theories in different logics cannot be identical. For simulation it suffices that conclusions are equal modulo tags. Given logics L_1 and L_2, with principal tags d_1 and d_2, respectively, we say two conclusions α in L_1 and β in L_2 are *equal modulo tags* if α is $+d_1 q$ and β is $+d_2 q$ or α is $-d_1 q$ and β is $-d_2 q$.

Thus we have the following definition of simulation and relative expressiveness. For more discussion on the motivations for the definitions, see (Maher 2012).

Definition 2

Let C be a class of defeasible theories.

We say D_1 in logic L_1 is *simulated* by D_2 in L_2 with respect to a class C if, for every modular addition A in C, $D_1 + A$ and $D_2 + A$ have the same conclusions in $\Sigma(D_1 + A)$, modulo tags.

We say a logic L_1 can be simulated by a logic L_2 with respect to a class C if every theory in L_1 can be simulated by some theory in L_2 with respect to additions from C.

We say L_2 is *more (or equal) expressive than* L_1 if L_1 can be simulated by L_2 with respect C.

Different notions of relative expressiveness arise from different choices for C. There were two classes of additions investigated in Maher (2012): the addition of *facts* (that is, A has the form $(F, \emptyset, \emptyset)$), and the addition of *rules* (that is, A has the form $(\emptyset, R, \emptyset)$). Simulation with respect to addition of rules is stronger than simulation with respect to addition of facts because any fact can equally be expressed as a strict rule with an empty body. We might also consider arbitrary additions, where A can be any defeasible theory.

The main results of Maher (2012) are that:

- $DL(\partial)$ and $DL(\partial^*)$ have equal expressiveness, with respect to addition of *facts*, as do $DL(\delta)$ and $DL(\delta^*)$

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

- neither $DL(\partial)$ nor $DL(\partial^*)$ is more expressive than $DL(\delta)$ or $DL(\delta^*)$, and vice versa, with respect to addition of *rules*
- when arbitrary additions are permitted, of the four defeasible logics under consideration, none is more expressive than any other

4 Blocked ambiguity simulates propagated ambiguity

We now show that every theory over an ambiguity propagating logic can be simulated by a theory over the corresponding ambiguity blocking logic. To begin, we show that $\mathbf{DL}(\partial^*)$ can simulate $\mathbf{DL}(\delta^*)$. Any defeasible theory D is transformed into a new theory. The new theory employs new propositions $strict(q)$ and $supp(q)$, for each literal q, and $supp_body(r)$, $comp(r)$, and $o(r)$, for each rule r. The new theory also introduces labels $p_d(r)$, $n_d(r,s)$, $p_s(r)$, $n_s(r,s)$, for each pair r,s of opposing rules in D. These are families of propositions and labels, not predicates, despite the notation.

Definition 3
Let $D = (F, R, >)$ be a defeasible theory with language Σ. We define the transformation T of D to $T(D) = (F', R', >')$ as follows:

1. The facts of $T(D)$ are the facts of D. That is, $F' = F$.
2. Every strict rule of R is included in R'.
3. For every literal q, R' contains

$$\begin{array}{lrcl} str(q): & q & \rightarrow & strict(q) \\ nstr(q): & & \Rightarrow & \neg strict(q) \end{array}$$

   and the superiority relation contains $nstr(q) >' str(q)$, for every q.
4. For each literal q in Σ, R' contains

$$q \Rightarrow supp(q)$$

5. For each strict or defeasible rule r of the form $b_1, \ldots, b_n \hookrightarrow_r q$ in R, R' contains

$$\begin{array}{rcl} supp(b_1), \ldots, supp(b_n) & \Rightarrow & supp_body(r) \\ supp_body(r), \neg o(r) & \Rightarrow & supp(q) \end{array}$$

   and, further, for each rule $s = B_s \hookrightarrow_s \sim q$ for $\sim q$ in R, where $s > r$, R' contains

$$\begin{array}{lrcl} n_s(r,s): & B_s & \Rightarrow & o(r) \\ p_s(r): & & \Rightarrow & \neg o(r) \end{array}$$

   and the superiority relation contains $n_s(s,r) >' p_s(s)$.
6. For each strict or defeasible rule $r = B_r \hookrightarrow_r q$ in R, R' contains

$$inf(r): \quad B_r, \neg comp(r), \neg strict(\sim q) \quad \Rightarrow \quad q$$

   and, further, for each rule $s = B_s \hookrightarrow_s \sim q$ for $\sim q$ in R, where $s \not< r$, R' contains

$$\begin{array}{lrcl} n_d(r,s): & supp_body(s) & \Rightarrow & comp(r) \\ p_d(r): & & \Rightarrow & \neg comp(r) \end{array}$$

   and the superiority relation contains $n_d(r,s) >' p_d(r)$.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

Parts 1 and 2 of the transformation preserve all the strict inferences from D. Part 3 allows us to distinguish strict conclusions from defeasible conclusions. The structure of these rules – where $str(q)$ is strict, $nstr(q)$ is defeasible, and $nstr(q) > str(q)$ – ensures that $strict(q)$ is inferred defeasibly iff q is inferred strictly, and $strict(q)$ fails iff strict inference of q fails. A similar structure of rules was previously used in Maher (2012) in showing that $\mathbf{DL}(\partial^*)$ can simulate $\mathbf{DL}(\partial)$ wrt addition of facts.

We use the proposition $supp(q)$ to indicate that the literal q is supported (i.e. $+\sigma^* q$ can be inferred), while the literal q refers to defeasible provability (wrt δ^*). Part 4 ensures that every literal that holds defeasibly is also supported. This property is justified by the inclusion theorem of Billington *et al.* (2010). Part 5 encodes the inference rules for support (i.e. σ^*). $supp_body(r)$ indicates that all literals in the body of rule r are supported. The head q of a rule r is supported if the body of r is supported and r is not overruled (i.e. all rules s that are superior to r fail). The overruling of r is indicated by $o(r)$. The rules $n_s(r, s)$ and $p_s(r)$ and the superiority relation ensure that $\neg o(r)$ is derived defeasibly iff there is no overruling rule s.

Rules $inf(r)$ in part 6 encode the inference rules for δ^*. q holds defeasibly iff the body of a rule r for q holds defeasibly and r has no competing rules (i.e. all rules for $\sim q$ not inferior to r have a body that fails wrt σ^*). The rules $n_d(r, s)$ and $p_d(r)$ and the superiority relation ensure that $\neg comp(r)$ is derived defeasibly iff there is no competing rule.

In this translation, the superiority relation in D is not directly represented by the superiority relation in $T(D)$. Instead, the superiority relation in D is used to restrict the instantiation of rules in the transformation, while the superiority relation in $T(D)$ is used to ensure that $o(r)$ and $\neg o(r)$ do not both fail, and similarly for $comp(r)$.

Example 4

To see the operation of this transformation, consider the following theory D, which demonstrates the difference between ambiguity propagation and blocking logics.

$$\begin{array}{ll} r_1 : \Rightarrow p & r_3 : \neg p \Rightarrow \neg q \\ r_2 : \Rightarrow \neg p & r_4 : \Rightarrow q \end{array}$$

In $\mathbf{DL}(\partial^*)$ from D we conclude $-\partial^* p$ and $-\partial^* \neg p$, $+\partial^* q$ and $-\partial^* \neg q$. In $\mathbf{DL}(\delta^*)$ from D we conclude $-\delta^* p$ and $-\delta^* \neg p$, $-\delta^* q$ and $-\delta^* \neg q$. We also conclude $+\sigma^* p$ and $+\sigma^* \neg p$, $+\sigma^* q$ and $+\sigma^* \neg q$.

$T(D)$ contains the following rules.

$$\begin{array}{rl} \Rightarrow supp_body(r_1) & supp_body(r_1), \neg o(r_1) \Rightarrow supp(p) \\ \Rightarrow supp_body(r_2) & supp_body(r_2), \neg o(r_2) \Rightarrow supp(\neg p) \\ supp(\neg p) \Rightarrow supp_body(r_3) & supp_body(r_3), \neg o(r_3) \Rightarrow supp(\neg q) \\ \Rightarrow supp_body(r_4) & supp_body(r_4), \neg o(r_4) \Rightarrow supp(q) \end{array}$$

$$\begin{array}{rl} p \Rightarrow supp(p) & p_s(r_1) : \Rightarrow \neg o(r_1) \\ \neg p \Rightarrow supp(\neg p) & p_s(r_2) : \Rightarrow \neg o(r_2) \\ q \Rightarrow supp(q) & p_s(r_3) : \Rightarrow \neg o(r_3) \\ \neg q \Rightarrow supp(\neg q) & p_s(r_4) : \Rightarrow \neg o(r_4) \end{array}$$

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

$$
\begin{aligned}
&inf(r_1): && \neg comp(r_1), \neg strict(\neg p) \Rightarrow && p\\
&inf(r_2): && \neg comp(r_2), \neg strict(p) \Rightarrow && \neg p\\
&inf(r_3): && \neg p, \neg comp(r_3), \neg strict(q) \Rightarrow && \neg q\\
&inf(r_4): && \neg comp(r_4), \neg strict(\neg q) \Rightarrow && q
\end{aligned}
$$

$$
\begin{aligned}
&p_d(r_1): && \Rightarrow \neg comp(r_1) && \qquad && n_d(r_1, r_2): && supp_body(r_2) \Rightarrow comp(r_1)\\
&p_d(r_2): && \Rightarrow \neg comp(r_2) && \qquad && n_d(r_2, r_1): && supp_body(r_1) \Rightarrow comp(r_2)\\
&p_d(r_3): && \Rightarrow \neg comp(r_3) && \qquad && n_d(r_3, r_4): && supp_body(r_4) \Rightarrow comp(r_3)\\
&p_d(r_4): && \Rightarrow \neg comp(r_4) && \qquad && n_d(r_4, r_3): && supp_body(r_3) \Rightarrow comp(r_4)
\end{aligned}
$$

$T(D)$ also contains the following superiority statements.

$$
\begin{aligned}
&nstr(p) >' str(p) && \qquad && n_d(r_1, r_2) >' p_d(r_1)\\
&nstr(\neg p) >' str(\neg p) && \qquad && n_d(r_2, r_1) >' p_d(r_2)\\
&nstr(q) >' str(q) && \qquad && n_d(r_3, r_4) >' p_d(r_3)\\
&nstr(\neg q) >' str(\neg q) && \qquad && n_d(r_4, r_3) >' p_d(r_4)
\end{aligned}
$$

There are some points to highlight in this example. Rules for *strict* and $\neg strict$ are omitted from the listing above because they are not of interest (D has no strict rules or facts); we will have conclusions $+\partial^* \neg strict(l)$ and $-\partial^* strict(l)$, for every literal l. There are no rules $n_s(r, s)$ in $T(D)$ because they only occur when $s > r$, and the superiority relation in D is empty. Consequently, there are no superiority statements of the form $n_s(r, s) >' p_s(r)$. It also follows that $+\partial^* \neg o(r)$ is concluded, for each rule r, and hence we can infer $+\partial^* supp(l)$, for each literal l except $\neg q$, reflecting the fact that these literals are supported in D, and $+\partial^* supp_body(r)$, for each rule r. We can then infer also $+\partial^* supp(\neg q)$. It then follows that $-\partial^* \neg comp(r)$ is concluded, for each r, using the superiority relation. Then, as a consequence of the rules $inf(r)$, we find that all literals l fail to be inferred (i.e. we conclude $-\partial^* l$, for each literal l). This expresses the ambiguity propagating behaviour of $\mathbf{DL}(\delta^*)$ from within $\mathbf{DL}(\partial^*)$.

Theorem 5

The ambiguity blocking logics ($\mathbf{DL}(\partial)$ and $\mathbf{DL}(\partial^*)$) can simulate the ambiguity propagating logics ($\mathbf{DL}(\delta)$ and $\mathbf{DL}(\delta^*)$) with respect to addition of facts.

5 Propagated ambiguity simulates blocked ambiguity

We now show that every theory over an ambiguity blocking logic can be simulated by a theory over the corresponding ambiguity propagating logic. To begin, we simulate $\mathbf{DL}(\partial^*)$ by $\mathbf{DL}(\delta^*)$. Any defeasible theory D is transformed into a new theory $T(D)$. The new theory employs new propositions $strict(q)$ and $undefeated(q)$ for each literal q in Σ, and employs labels $str(q)$ and $nstr(q)$ for each literal q in Σ, and $n_d(r, s)$ and $p_d(r)$ for each pair of opposing rules r, s in R.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

Definition 6
Let $D = (F, R, >)$ be a defeasible theory with language Σ. We define the transformation T of D to $T(D) = (F', R', >')$ as follows:

1. The facts of $T(D)$ are the facts of D. That is, $F' = F$.
2. Every strict rule of R is included in R'.
3. For every literal q, R' contains

$$\begin{array}{lrcl} str(q): & q & \rightarrow & strict(q) \\ nstr(q): & & \Rightarrow & \neg strict(q) \\ t(q): & strict(q) & \Rightarrow & true(q) \\ nt(q): & & \Rightarrow & \neg true(q) \end{array}$$

   and the superiority relation contains $nstr(q) >' str(q)$ and $t(q) >' nt(q)$, for every q.
4. For each literal q, R' contains

$$undefeated(q) \Rightarrow q$$

   For each strict or defeasible rule $r = B_r \hookrightarrow_r q$ for q in R, R' contains

$$p_d(r): \quad B_r, \neg true(\sim q) \Rightarrow undefeated(q)$$

   and, further, for each rule $s = B_s \hookrightarrow_s \sim q$ for $\sim q$ in R, where $r \not> s$, R' contains

$$n_d(r, s): \quad B_s \Rightarrow \neg undefeated(q)$$

   and the superiority relation contains $n_d(r, s) >' p_d(r)$.

Parts 1 and 2 preserve all the strict inferences from D. Part 3 allows us to distinguish strict conclusions from defeasible conclusions. For this transformation – compared to the transformation in the previous section – extra rules t and nt are needed. These rules ensure that δ^* and σ^* agree on the literals $true(q)$, that is, from $T(D)+A$ we conclude $+\delta^* true(q)$ iff we conclude $+\sigma^* true(q)$ iff $D+A \vdash +\Delta q$. (See Lemma 18 in the appendix.) In comparison, we never infer $-\sigma^* \neg strict(q)$ and always infer $+\sigma^* \neg strict(q)$, independent of D.[1]

Part 4 encodes the inference rules for ∂^*: q holds defeasibly if the body of a rule r for q holds defeasibly, $\sim q$ is not established strictly, and r is not defeated (i.e. all rules not inferior to r have a body that fails wrt ∂^*). The requirement that r is not defeated is expressed through the use of rules $n_d(r, s)$ opposing $p_d(r)$ for each rule s in D not inferior to r. The rules $n_d(r, s)$ are superior to $p_d(r)$ in $T(D)$, thus ensuring that $undefeated(q)$ is inferred iff r is not defeated.

Example 7
To see the operation of this transformation, consider (again) the following theory D, which demonstrates the difference between ambiguity propagation and blocking logics.

[1] This also demonstrates a flaw in Maher (2012). In that paper, the transformation used to simulate **DL**(δ) with **DL**(δ^*) fails to use these extra rules, and thus is incorrect. Definition 13 has a corrected transformation.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

$$r_1 : \Rightarrow p \qquad r_3 : \neg p \Rightarrow \neg q$$
$$r_2 : \Rightarrow \neg p \qquad r_4 : \Rightarrow q$$

In $\mathbf{DL}(\partial^*)$ from D we conclude $-\partial^* p$ and $-\partial^* \neg p$, $+\partial^* q$ and $-\partial^* \neg q$. In $\mathbf{DL}(\delta^*)$ from D we conclude $-\delta^* p$ and $-\delta^* \neg p$, $-\delta^* q$ and $-\delta^* \neg q$. We also conclude $+\sigma^* p$ and $+\sigma^* \neg p$, $+\sigma^* q$ and $+\sigma^* \neg q$.

$T(D)$ contains the following rules and superiority relation.

$$n_d(r_1, r_2) : \Rightarrow \neg undefeated(p) \qquad p_d(r_1) : \neg true(\neg p) \Rightarrow undefeated(p)$$
$$n_d(r_2, r_1) : \Rightarrow \neg undefeated(\neg p) \qquad p_d(r_2) : \neg true(p) \Rightarrow undefeated(\neg p)$$
$$n_d(r_3, r_4) : \Rightarrow \neg undefeated(\neg q) \qquad p_d(r_3) : \neg p, \neg true(\neg q) \Rightarrow undefeated(\neg q)$$
$$n_d(r_4, r_3) : \neg p \Rightarrow \neg undefeated(q) \qquad p_d(r_4) : \neg true(q) \Rightarrow undefeated(q)$$

$$undefeated(p) \Rightarrow p$$
$$undefeated(\neg p) \Rightarrow \neg p$$
$$undefeated(q) \Rightarrow q$$
$$undefeated(\neg q) \Rightarrow \neg q$$

$$n_d(r_1, r_2) >' p_d(r_1) \qquad nstr(p) >' str(p)$$
$$n_d(r_2, r_1) >' p_d(r_2) \qquad nstr(\neg p) >' str(\neg p)$$
$$n_d(r_3, r_4) >' p_d(r_3) \qquad nstr(q) >' str(q)$$
$$n_d(r_4, r_3) >' p_d(r_4) \qquad nstr(\neg q) >' str(\neg q)$$

The rules concerning *strict* and *true* have been omitted. Because there are no facts or strict rules in D we will infer $-\delta^* strict(s)$, and hence $+\delta^* \neg true(s)$ and $+\sigma^* \neg true(s)$ for each literal $s \in \Sigma$. However, because of the superiority of n_d over p_d, we infer $-\sigma^* undefeated(\neg p)$ and $-\delta^* undefeated(\neg p)$ and hence $-\sigma^* \neg p$ and $-\delta^* \neg p$ (and similarly for p). Hence, the body of $p_d(r_3)$ fails, and we infer $-\delta^* \neg q$. Similarly, the body of $n_d(r_4, r_3)$ fails, and hence we infer $+\delta^* q$. This reflects the ambiguity blocking behaviour of $\mathbf{DL}(\partial^*)$ from within the ambiguity propagating logic $\mathbf{DL}(\delta^*)$.

The proof of correctness of this simulation is complicated by the fact that inference rules for δ^* and σ^* are defined mutually recursively, while the inference rules for ∂^* are directly recursive. This difference in structure makes a direct inductive proof difficult. The problem is resolved by a "tight" simulating transformation that is able to simulate $\mathbf{DL}(\partial^*)$ (wrt addition of facts) in any of the **DL** logics.

Theorem 8
For $d \in \{\delta, \delta^*, \partial\}$, $\mathbf{DL}(d)$ can simulate $\mathbf{DL}(\partial^*)$ with respect to addition of facts

Combining Theorems 5 and 8 with results from (Maher 2012), we see that all logics of the **DL** framework are equally expressive in terms of simulation wrt addition of facts.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

6 Simulation of individual defeat wrt addition of rules

The following definition defining $D' = (F', R', <')$ from D is repeated from (Maher 2012).

Definition 9
We add the following rules

1. The facts of D' are the facts of D. That is, $F' = F$.
2. For each rule $r = B \hookrightarrow_r q$ in R, R' contains

$$\begin{array}{lrcl} p(r): & B & \hookrightarrow_r & h(r) \\ s(r): & h(r) & \rightarrow & q \end{array}$$

   and, further, for each rule $r' = B' \hookrightarrow_{r'} \sim q$ for $\sim q$ in R, R' contains

$$n(r, r'): \quad B' \quad \hookrightarrow_{r'} \quad \neg h(r)$$

3. For every $r > r'$ in D, where r and r' are rules for opposite literals, D' contains $p(r) >' n(r, r')$ and $n(r', r) >' p(r')$.

It was shown in Maher (2012) that, using this transformation, $\mathbf{DL}(\partial)$ simulates $\mathbf{DL}(\partial^*)$ and $\mathbf{DL}(\delta)$ simulates $\mathbf{DL}(\delta^*)$, wrt addition of facts.

On the surface, it might appear that this result extends readily to addition wrt rules: since the added rules do not participate in the superiority relation of the combined theory, it might be expected that the difference between team defeat and individual defeat is irrelevant. However, that expectation is misleading. The following example shows that this transformation *does not* provide a simulation of $\mathbf{DL}(\partial^*)$ by $\mathbf{DL}(\partial)$ wrt addition of *rules*.

Example 10
Let D consist of the rules

$$\begin{array}{lcr} r_1: & \Rightarrow & p \\ r_2: & \Rightarrow & \neg p \end{array}$$

Then $T(D)$ consists of the following rules

$$\begin{array}{lrcl} p(r_1): & & \Rightarrow & h(r_1) \\ n(r_1, r_2): & & \Rightarrow & \neg h(r_1) \\ p(r_2): & & \Rightarrow & h(r_2) \\ n(r_2, r_1): & & \Rightarrow & \neg h(r_2) \\ s(r_1): & h(r_1) & \Rightarrow & p \\ s(r_2): & h(r_2) & \Rightarrow & \neg p \end{array}$$

Now, let A be the rule

$$\Rightarrow \quad p$$

Clearly, $D + A \vdash -\partial^* p$ (and $D + A \vdash -\partial^* \neg p$), since r_2 cannot be overruled. However, $T(D) + A \vdash -\partial h(r_2)$, since $n(r_2, r_1)$ cannot be overruled, and hence $s(r_2)$ fails. This leaves the rule for p in A without competition, and so $T(D) + A \vdash +\partial p$.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

A similar but more complex example (given in the appendix) shows the transformation also does not provide a simulation of $\mathbf{DL}(\delta^*)$ by $\mathbf{DL}(\delta)$ wrt addition of rules. These problems arise because if a rule body succeeds in D for each of q and $\sim q$, and the rules are not overruled, the simulation D' has all bodies for q and $\sim q$ failing. An applicable rule for q (or $\sim q$) in A thus has a competitor in D, but not in D'. In this way D' differs from D, and the examples show that addition of rules can make this difference observable.

To avoid these problems, we add extra rules to those in Definition 9.

Definition 11
We define $T(D)$ as the theory $(F', R', >')$ consisting of the facts, rules and superiority statements from D' in Definition 9, and the following.

4. For each literal q, R' contains
$$o(q) : \quad one(q) \quad \rightsquigarrow \quad q$$
5. For each rule $r = B_r \hookrightarrow_r q$ in R, R' contains
$$B_r \quad \Rightarrow \quad one(q)$$
6. For every rule r for q, $>'$ contains $s(r) >' o(\sim q)$.

Parts 4 and 5 of this definition introduce an additional rule for each literal $\sim q$ which, however, is subordinate to the methods to derive q in the original transformation in the sense that a derivation of q in the original transformation will overrule (part 6) a derivation of $\sim q$ using part 4. The rules in part 4 are defeaters, so they cannot be used to derive any conclusions.

The effect of the extended definition on Example 10 is to add the following to the transformed theory:

$$\begin{array}{llllll} \Rightarrow & one(p) & o(p) & : & one(p) & \rightsquigarrow \quad p \\ \Rightarrow & one(\neg p) & o(\neg p) & : & one(\neg p) & \rightsquigarrow \quad \neg p \end{array}$$

$$s(r_1) > o(\neg p) \qquad\qquad s(r_2) > o(p)$$

We now have $T(D) + A \vdash -\partial p$, since the rule $o(\neg p)$ provides a non-failed competitor to the rule in A. More generally, we find that, through the extended transformation, team defeat logics can simulate the corresponding individual defeat logics with respect to addition of rules.

Theorem 12
The logic $\mathbf{DL}(\partial)$ can simulate $\mathbf{DL}(\partial^*)$ with respect to addition of rules.
The logic $\mathbf{DL}(\delta)$ can simulate $\mathbf{DL}(\delta^*)$ with respect to addition of rules.

7 Simulation of team defeat wrt addition of rules

The same theory D and addition A as in Example 10 demonstrates that the simulation of $\mathbf{DL}(\partial)$ by $\mathbf{DL}(\partial^*)$ wrt addition of facts exhibited in Maher (2012) does not extend to addition of rules.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

The transformation below modifies the one of Maher (2012) by treating strict rules differently (following Definition 6), adding a competitor for each literal q (following Definition 11), and employing separate defeasible rules to accommodate differences between the δ and σ inference rules. We use a construction to restrict one class of defeasible rules to use only in simulating σ inference; it is not necessary to restrict the other class because $\delta \subseteq \sigma$, by the inclusion theorem.

Definition 13

We define the transformation T of D to $T(D) = (F', R', >')$ as follows:

1. The facts of $T(D)$ are the facts of D. That is, $F' = F$.
2. Every strict rule of R is included in R'.
3. For every literal q, R' contains

$$\begin{array}{lrcl} str(q): & q & \rightarrow & strict(q) \\ nstr(q): & & \Rightarrow & \neg strict(q) \\ t(q): & strict(q) & \Rightarrow & true(q) \\ nt(q): & & \Rightarrow & \neg true(q) \end{array}$$

   and the superiority relation contains $nstr(q) >' str(q)$ and $t(q) >' nt(q)$, for every q.
4. For each ordered pair of opposing rules $r_i = (B_i \hookrightarrow_i \sim q)$ and $r_j = (B_j \hookrightarrow_j q)$ in R, where r_j is not a defeater, R' contains

$$\begin{array}{lrcl} R1_{ij}: & B_i & \hookrightarrow_i \neg & d(r_i, r_j) \\ R2_{ij}: & B_j & \Rightarrow & d(r_i, r_j) \\ R3_{ij}: & true(q) & \Rightarrow & d(r_i, r_j) \\ & d(r_i, r_j) & \Rightarrow & d(r_i) \\ & fail(r_i) & \Rightarrow & d(r_i) \\ NF_i: & B_i & \Rightarrow \neg & fail(r_i) \\ F_i: & & \Rightarrow & fail(r_i) \end{array}$$

   and $R2_{ij} >' R1_{ij}$ iff $r_j > r_i$, $R3_{ij} >' R1_{ij}$ for every i and j, and $NF_i > F_i$ for every i.

   If there is no strict or defeasible rule r_j for q in D then only the last three rules appear in R', for each i.
5. For each literal q, and each strict or defeasible rule $r = (B_r \hookrightarrow_r q)$ in R, R' contains

$$B_r \Rightarrow one(q)$$

6. For each literal q, R' contains

$$s(q): \quad one(q), \neg true(\sim q), d(s_1), \ldots, d(s_k) \Rightarrow q$$

   where $s_1, \ldots, s_k$ are the rules for $\sim q$
7. For each literal q and for each strict or defeasible rule r for q, R' contains

$$supp(q): \quad B_r, d_\sigma(s_1, r), \ldots, d_\sigma(s_k, r), g, \neg g \Rightarrow q$$

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

where B_r is the body of r, $s_1, \ldots, s_k$ are the rules for $\sim q$, and for every strict or defeasible rule r and opposing rule s, R' contains

$$\begin{array}{lrcl} a(s,r): & B_s & \Rightarrow & \neg d_\sigma(s,r) \\ b(s,r): & B_r & \Rightarrow & d_\sigma(s,r) \end{array}$$

The superiority relation contains $a(s,r) > b(s,r)$ iff $s > r$. R' also contains the rules

$$\begin{array}{rcl} \Rightarrow & & g \\ \Rightarrow & & \neg g \end{array}$$

8. For each rule $r = B_r \hookrightarrow_r q$ in R, R' contains

$$B_r \Rightarrow o(q)$$

9. For each literal q, R' contains

$$o(q): \quad o(q) \rightsquigarrow q$$

and $>'$ contains $s(q) >' o(\sim q)$.

Parts 1–3 allow us to characterize strict conclusions. Part 4 expresses whether a rule is defeated or not, while part 6 expresses that q can be concluded if there is an applicable strict or defeasible rule for q, all attempts to strictly derive $\sim q$ fail finitely, and all opposing rules are defeated. While this expresses properly the inference rules for ∂ and δ, the inference rule for σ omits the condition on strict derivation of $\sim q$ and has a slightly different form of defeat. We need part 7 to express inference (and defeat) for σ. g and $\neg g$ are used to restrict the applicability of this rule to σ^*; we have $T(D)+A \vdash +\sigma^* g$, but $T(D)+A \vdash -\partial^* g$ and $T(D)+A \vdash -\delta^* g$ (and the same for $\neg g$). Parts 8 and 9 redress the lack of a competitor in the same way as in Definition 11.

Theorem 14
The logic **DL**(∂^*) can simulate **DL**(∂) with respect to addition of rules.
The logic **DL**(δ^*) can simulate **DL**(δ) with respect to addition of rules.

8 Conclusions

We have shown that the logics of the **DL** framework are equally expressive when relative expressiveness is formulated as ability to simulate in the presence of additional facts. This involved the introduction of two new transformations simulating, respectively, a logic that blocks ambiguity and a logic that propagates ambiguity.

We also completed the study of relative expressiveness wrt addition of rules. Figure 1 shows this relation on the logics in **DL**, where an arrow from d_1 to d_2 expresses that $DL(d_1)$ can be simulated by $DL(d_2)$ with respect to the addition of rules. <> between tags expresses that the two corresponding logics have incomparable expressiveness. It is clear that **DL** breaks into two classes of logics of different expressiveness.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press

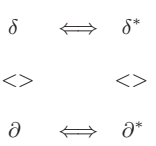


Fig. 1. Relative expressiveness of logics in **DL** using simulation wrt addition of rules.

While the issue of relative expressiveness within the framework **DL** is now largely resolved, this same approach can be applied to relate these logics to other logics. We can expect the same results for the **WFDL** logics (Maher and Governatori 1999; Maher *et al.* 2011), because of their similarity to **DL**, but their relation to the defeasible logics of Maier and Nute (2006; 2010) will be of interest. Even more interesting will be to address other systems of defeasible reasoning, such as argumentation (Dung 1995; Rahwan and Simari 2009).

Acknowledgements

The author thanks the referees for their comments, and UNSW, Canberra for a grant supporting this work.

References

Antoniou, G., Billington, D., Governatori, G. and Maher, M. J. 2000. A flexible framework for defeasible logics. In *AAAI/IAAI*, AAAI Press / The MIT Press, 405–410.

Billington, D., Antoniou, G., Governatori, G. and Maher, M. J. 2010. An inclusion theorem for defeasible logics. *ACM Transactions on Computational Logic 12,* 1, 6.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence 77,* 2, 321–358.

Maher, M. J. 2001. Propositional defeasible logic has linear complexity. *TPLP 1,* 6, 691–711.

Maher, M. J. 2012. Relative expressiveness of defeasible logics. *TPLP 12,* 4-5, 793–810.

Maher, M. J. and Governatori, G. 1999. A semantic decomposition of defeasible logics. In *AAAI/IAAI*, AAAI Press, 299–305.

Maher, M. J., Governatori, G. and Lam, H. P. 2011. Well-founded defeasible logics. Tech. rep.

Maier, F. and Nute, D. 2006. Ambiguity propagating defeasible logic and the well-founded semantics. In *JELIA*, M. Fisher, W. van der Hoek, B. Konev and A. Lisitsa, Eds. Lecture Notes in Computer Science, vol. 4160. Springer, 306–318.

Maier, F. and Nute, D. 2010. Well-founded semantics for defeasible logic. *Synthese 176,* 2, 243–274.

Rahwan, I. and Simari, G. 2009. *Argumentation in Artificial Intelligence*. Springer.

https://doi.org/10.1017/S1471068413000367 Published online by Cambridge University Press