Online appendix for the paper

Annotated Defeasible Logic

published in Theory and Practice of Logic Programming

Guido Governatori
Data61, CSIRO, Australia
E-mail: guido.governatori@data61.csiro.au

Michael J. Maher
Reasoning Research Institute, Australia
E-mail: michael.maher@reasoning.org.au

submitted 2 May 2017; revised 20 June 2017; accepted 5 July 2017

Inference Rules

Defeasible logics are usually defined via their proof mechanism. Here we present the inference rules for the four defeasible logics we integrate within annotated defeasible logic. Each inference rule is labelled by the kind of conclusions it infers. The presentation is adapted from (Billington et al. 2010). A defeasible logic is determined by the inference rules it allows. For example, $\mathbf{DL}(\partial)$ allows $+\partial$ and $-\partial$, while $\mathbf{DL}(\delta)$ allows $+\delta$, $-\delta$, $+\sigma_\delta$, and $-\sigma_\delta$.

A proof P is a sequence of conclusions. The conclusion at position i in the sequence is denoted by $P(i)$, and a prefix of the proof of length i is denoted by $P[1..i]$. The inference rules establish when a conclusion can be drawn at position $i+1$, given the conclusions already proved ($P[1..i]$). Where q is a literal, $R_{sd}[q]$ denotes the set of strict or defeasible rules with head q, while $R[q]$ denotes the set of all rules (including defeaters) with head q. For a rule r, $A(r)$ denotes the antecedent (or body) of r.

$+\partial$) Infer $P(i+1) = +\partial q$ if either
- .1) $+\Delta q \in P[1..i]$; or
- .2) The following three conditions all hold.
  - .1) $\exists r \in R_{sd}[q]\ \forall a \in A(r), +\partial a \in P[1..i]$, and
  - .2) $-\Delta{\sim}q \in P[1..i]$, and
  - .3) $\forall s \in R[{\sim}q]$ either
    - .1) $\exists a \in A(s), -\partial a \in P[1..i]$; or
    - .2) $\exists t \in R_{sd}[q]$ such that
      - .1) $\forall a \in A(t), +\partial a \in P[1..i]$, and
      - .2) $t > s$.

$-\partial$) Infer $P(i+1) = -\partial q$ if
- .1) $-\Delta q \in P[1..i]$, and
- .2) either
  - .1) $\forall r \in R_{sd}[q]\ \exists a \in A(r), -\partial a \in P[1..i]$; or
  - .2) $+\Delta{\sim}q \in P[1..i]$; or
  - .3) $\exists s \in R[{\sim}q]$ such that
    - .1) $\forall a \in A(s), +\partial a \in P[1..i]$, and
    - .2) $\forall t \in R_{sd}[q]$ either
      - .1) $\exists a \in A(t), -\partial a \in P[1..i]$; or
      - .2) not$(t > s)$.

$+\delta$) Infer $P(i+1) = +\delta q$ if either
.1) $+\Delta q \in P[1..i]$; or
.2) The following three conditions all hold.
.1) $\exists r \in R_{sd}[q]\ \forall a \in A(r), +\delta a \in P[1..i]$, and
.2) $-\Delta{\sim}q \in P[1..i]$, and
.3) $\forall s \in R[{\sim}q]$ either
.1) $\exists a \in A(s), -\sigma_\delta a \in P[1..i]$; or
.2) $\exists t \in R_{sd}[q]$ such that
.1) $\forall a \in A(t), +\delta a \in P[1..i]$, and
.2) $t > s$.

$-\delta$) Infer $P(i+1) = -\delta q$ if
.1) $-\Delta q \in P[1..i]$, and
.2) either
.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r), -\delta a \in P[1..i]$; or
.2) $+\Delta{\sim}q \in P[1..i]$; or
.3) $\exists s \in R[{\sim}q]$ such that
.1) $\forall a \in A(s), +\sigma_\delta a \in P[1..i]$, and
.2) $\forall t \in R_{sd}[q]$ either
.1) $\exists a \in A(t), -\delta a \in P[1..i]$; or
.2) not$(t > s)$.

$+\sigma_\delta$) Infer $P(i+1) = +\sigma_\delta q$ if either
.1) $+\Delta q \in P[1..i]$; or
.2) $\exists r \in R_{sd}[q]$ such that
.1) $\forall a \in A(r), +\sigma_\delta a \in P[1..i]$, and
.2) $\forall s \in R[{\sim}q]$ either
.1) $\exists a \in A(s), -\delta a \in P[1..i]$; or
.2) not$(s > r)$.

$-\sigma_\delta$) Infer $P(i+1) = -\sigma_\delta q$ if
.1) $-\Delta q \in P[1..i]$, and
.2) $\forall r \in R_{sd}[q]$ either
.1) $\exists a \in A(r), -\sigma_\delta a \in P[1..i]$; or
.2) $\exists s \in R[{\sim}q]$ such that
.1) $\forall a \in A(s), +\delta a \in P[1..i]$, and
.2) $s > r$.

$+\partial^*$) Infer $P(i+1) = +\partial^* q$ if either
.1) $+\Delta q \in P[1..i]$; or
.2) $\exists r \in R_{sd}[q]$ such that
.1) $\forall a \in A(r), +\partial^* a \in P[1..i]$, and
.2) $-\Delta{\sim}q \in P[1..i]$, and
.3) $\forall s \in R[{\sim}q]$ either
.1) $\exists a \in A(s), -\partial^* a \in P[1..i]$; or
.2) $r > s$.

$-\partial^*$) Infer $P(i+1) = -\partial^* q$ if
.1) $-\Delta q \in P[1..i]$, and
.2) $\forall r \in R_{sd}[q]$ either
.1) $\exists a \in A(r), -\partial^* a \in P[1..i]$; or
.2) $+\Delta{\sim}q \in P[1..i]$; or
.3) $\exists s \in R[{\sim}q]$ such that
.1) $\forall a \in A(s), +\partial^* a \in P[1..i]$, and
.2) not$(r > s)$.

$+\delta^*$) Infer $P(i+1) = +\delta^* q$ if either
.1) $+\Delta q \in P[1..i]$; or
.2) $\exists r \in R_{sd}[q]$ such that
.1) $\forall a \in A(r), +\delta^* a \in P[1..i]$, and
.2) $-\Delta{\sim}q \in P[1..i]$, and
.3) $\forall s \in R[{\sim}q]$ either
.1) $\exists a \in A(s), -\sigma^*_{\delta^*} a \in P[1..i]$; or
.2) $r > s$.

$-\delta^*$) Infer $P(i+1) = -\delta^* q$ if
.1) $-\Delta q \in P[1..i]$, and
.2) $\forall r \in R_{sd}[q]$ either
.1) $\exists a \in A(r), -\delta^* a \in P[1..i]$; or
.2) $+\Delta{\sim}q \in P[1..i]$; or
.3) $\exists s \in R[{\sim}q]$ such that
.1) $\forall a \in A(s), +\sigma^*_{\delta^*} a \in P[1..i]$, and
.2) not$(r > s)$.

$+\sigma^*_{\delta^*}$) Infer $P(i+1) = +\sigma^*_{\delta^*} q$ if either
.1) $+\Delta q \in P[1..i]$; or
.2) $\exists r \in R_{sd}[q]$ such that
.1) $\forall a \in A(r), +\sigma^*_{\delta^*} a \in P[1..i]$, and
.2) $\forall s \in R[{\sim}q]$ either
.1) $\exists a \in A(s), -\delta^* a \in P[1..i]$; or
.2) not$(s > r)$.

$-\sigma^*_{\delta^*}$) Infer $P(i+1) = -\sigma^*_{\delta^*} q$ if
.1) $-\Delta q \in P[1..i]$, and
.2) $\forall r \in R_{sd}[q]$ either
.1) $\exists a \in A(r), -\sigma^*_{\delta^*} a \in P[1..i]$; or
.2) $\exists s \in R[{\sim}q]$ such that
.1) $\forall a \in A(s), +\delta^* a \in P[1..i]$, and
.2) $s > r$.

Original Meta-programs

The original metaprograms (Maher and Governatori 1999; Antoniou et al. 2000) for the four main forms of defeasibility are outlined below. They consist of clauses $c1$ and $c2$, defining `definitely`, clauses defining `rule` and `supportive_rule` (see body of the paper), and a selection of the following clauses for each form of defeasibility.

$c21$ `defeasibly`(X) :-
  `definitely`(X).

$c22$ `defeasibly`(X) :-
  `not definitely`$(\sim X)$,
  `supportive_rule`$(R, X, [Y_1, \ldots, Y_n])$,
  `defeasibly`(Y_1),...,`defeasibly`(Y_n),
  `not overruled`(R, X).

$c23$ `overruled`(R, X) :-
  `rule`$(S, \sim X, [U_1, \ldots, U_n])$,
  `defeasibly`(U_1),...,`defeasibly`(U_n),
  `not defeated`$(S, \sim X)$.

$c24$ `defeated`$(S, \sim X)$:-
  `sup`(T, S),
  `supportive_rule`$(T, X, [V_1, \ldots, V_n])$,
  `defeasibly`(V_1),...,`defeasibly`(V_n).

$c25$ `supported`(X) :-
  `definitely`(X).

$c26$ `supported`(X) :-
  `supportive_rule`$(R, X, [Y_1, \ldots, Y_n])$,
  `supported`(Y_1),...,`supported`(Y_n),
  `not beaten`(R, X).

$c27$ `beaten`(R, X) :-
  `rule`$(S, \sim X, [W_1, \ldots, W_n])$,
  `defeasibly`(W_1),...,`defeasibly`(W_n),
  `sup`(S, R).

$c28$ `overruled`(R, X) :-
  `rule`$(S, \sim X, [U_1, \ldots, U_n])$,
  `supported`(U_1),...,`supported`(U_n),
  `not defeated`$(S, \sim X)$.

$c29$ `overruled`(R, X) :-
  `rule`$(S, \sim X, [U_1, \ldots, U_n])$,
  `defeasibly`(U_1),...,`defeasibly`(U_n),
  `not sup`(R, S).

$c30$ `overruled`(R, X) :-
  `rule`$(S, \sim X, [U_1, \ldots, U_n])$,
  `supported`(U_1),...,`supported`(U_n),
  `not sup`(R, S).

The selection of clauses for each meta-program is as follows:
$\mathcal{M}_\partial$ contains the clauses $c21$ - $c24$.
$\mathcal{M}_\delta$ contains the clauses $c21$ - $c22$, $c28$, $c24$, and $c25$ - $c27$.
$\mathcal{M}_{\partial^*}$ contains the clauses $c21$ - $c22$, and $c29$.
$\mathcal{M}_{\delta^*}$ consists of the clauses $c21$ - $c22$, $c30$, and $c25$ - $c27$.

Proofs of results

We present (sketches of) proofs for the results in the paper.

Theorem 1
Let $D = (F, R, >)$ be a defeasible theory, and α be an annotation function for that theory. Let $d \in \{\delta^*, \delta, \partial^*, \partial\}$.

Suppose $\alpha(R)$ contains only annotations `free` and d, and there is no fail-expression in R. Then, for every literal q

- $\mathcal{M}(\alpha(D)) \models_K \texttt{defeasibly}_d(d\ q)$ iff $\mathcal{M}_d(D) \models_K \texttt{defeasibly}(q)$
- $\mathcal{M}(\alpha(D)) \models_K \neg\texttt{defeasibly}_d(d\ q)$ iff $\mathcal{M}_d(D) \models_K \neg\texttt{defeasibly}(q)$

Furthermore, if $d \in \{\delta^*, \delta\}$,

- $\mathcal{M}(\alpha(D)) \models_K \texttt{supported}_d(d\ q)$ iff $\mathcal{M}_d(D) \models_K \texttt{supported}(q)$
- $\mathcal{M}(\alpha(D)) \models_K \neg\texttt{supported}_d(d\ q)$ iff $\mathcal{M}_d(D) \models_K \neg\texttt{supported}(q)$

Proof
(Sketch) The proof of this theorem is similar for each tag d. For brevity, we only provide the details for δ. The proof is based on unfolding $\mathcal{M}(\alpha(D))$ until it has essentially the same form as an unfolding of $\mathcal{M}_d(D)$. The form of unfolding we use uses clauses from the current program, and may be applied as long as no clause is used to unfold an atom in its own body. Such unfolding preserves the Kunen semantics (i.e. 3-valued models of the Clark-completion) of a logic program by essentially the same argument that it preserves the 2-valued models (Maher 1988). Clauses $c1$, $c2$, and $c5$ are the same in both $\mathcal{M}$ and $\mathcal{M}_d$, so we will essentially ignore them.

In both $\mathcal{M}(\alpha(D))$ and $\mathcal{M}_\delta(D)$ we unfold all occurrences of the predicates used to represent the annotated defeasible theory, and `rule` and `supportive_rule`. Then, in $\mathcal{M}(\alpha(D))$, we unfold all occurrences of the predicates specifying the type of each tag: `ambiguity_propagating`, `ambiguity_blocking`, `proof_tag`, `team`, and `indiv`. At this point clauses derived from $c6$-$c8$ are ground, while clauses derived from $c10$ only have a single, unused variable X in their heads. Similarly, clauses derived from $c14$ and $c15$ are ground.

Then unfold all $\texttt{defeasibly}_\texttt{Z}(\texttt{free}\ L)$ atoms. This will not result in a clause unfolding itself: in $c3$ because Z is not `free`, and in $c6$ because Y is not `free`. Similarly, we unfold all $\texttt{supported}_\texttt{Z}(\texttt{free}\ L)$ atoms. As a result, `free` only occurs in the head of clauses derived from $c3$ and $c11$.

Finally, unfold all $\mathcal{M}(\alpha(D))$ and `defeated` atoms in $\mathcal{M}_\delta(D)$. At this stage, clauses derived from $\mathcal{M}_\delta(D)$ are essentially the same as some of the clauses derived from $\mathcal{M}(\alpha(D))$ with subscript δ; the differences are in the name/arity of predicates (e.g., `defeasibly` versus $\texttt{defeasibly}_\delta$) and the presence of rules with heads of the form $\texttt{defeasibly}_\delta(\texttt{free}\ L)$, $\texttt{supported}_\delta(\texttt{free}\ L)$, $\texttt{defeasibly}_\delta(\texttt{fail}\ L)$ or $\texttt{supported}_\delta(\texttt{fail}\ L)$. However, no atom with subscript δ depends on a predicate with a different subscript, nor on clauses with `free` or `fail` in the head. Hence, the consequences of $\mathcal{M}(D)$ of the form $\texttt{defeasibly}_\delta(q)$ and $\texttt{supported}_\delta(q)$ are unaffected by the presence or absence of such rules, and so we delete them all.

Consequently, the two transformed programs are the same (modulo predicate renaming), and hence have the same conclusions. Since the transformations preserve the semantics of the programs, the result follows. □

Corollary 2
Suppose that α_F is the free annotation function for D. Let $d \in \{\delta^*, \delta, \partial^*, \partial\}$. Then, for every literal q,

- $\mathcal{M}(\alpha_F(D)) \models_K \texttt{defeasibly}_{\texttt{d}}(q)$ iff $D \vdash +dq$
- $\mathcal{M}(\alpha_F(D)) \models_K \neg\texttt{defeasibly}_{\texttt{d}}(q)$ iff $D \vdash -dq$

Furthermore, if $d \in \{\delta^*, \delta\}$,

- $\mathcal{M}(\alpha_F(D)) \models_K \texttt{supported}_{\texttt{d}}(q)$ iff $D \vdash +\sigma_d q$
- $\mathcal{M}(\alpha_F(D)) \models_K \neg\texttt{supported}_{\texttt{d}}(q)$ iff $D \vdash -\sigma_d q$

Proof
The corollary follows from applying the previous theorem for each tag d to the case where α is the free annotation function, and the correctness of the individual meta-programs. □

Theorem 3 (*Inclusion Theorem*)
Let D be an annotated defeasible theory.

(a) $+\Delta(D) \subseteq +\delta^*(D) \subseteq +\delta(D) \subseteq +\partial(D) \subseteq +\sigma_\delta(D) \subseteq +\sigma_{\delta^*}(D)$
(b) $-\sigma_{\delta^*}(D) \subseteq -\sigma_\delta(D) \subseteq -\partial(D) \subseteq -\delta(D) \subseteq -\delta^*(D) \subseteq -\Delta(D)$
(c) $+\partial(D) \subseteq +\sigma_\partial(D) \subseteq +\sigma_\delta(D)$
(d) $-\sigma_\delta(D) \subseteq -\sigma_\partial(D) \subseteq -\partial(D)$
(e) $+\delta^*(D) \subseteq +\partial^*(D) \subseteq +\sigma_{\partial^*}(D) \subseteq +\sigma_{\delta^*}(D)$
(f) $-\sigma_{\delta^*}(D) \subseteq -\sigma_{\partial^*}(D) \subseteq -\partial^*(D) \subseteq -\delta^*(D)$

Proof
(Sketch) Let $\Phi = \Phi_{\mathcal{M}(D)}$ be Fitting's semantic function for the logic program $\mathcal{M}(D)$ (Fitting 1985). Recall that Kunen's semantics is the set of all consequences of $\Phi \uparrow n$ for any finite n. We prove the containments by induction on the iteration of Φ. For brevity, we omit parts of the induction hypothesis related to proving (a). We also omit the parts related to (b), (d) and (f) since, by the Principle of Strong Negation (Antoniou et al. 2000), their statements and proof are symmetric to those for the positive conclusions. The induction hypothesis contains

$$
\begin{array}{rcll}
\texttt{supported}_{\partial^*} \subseteq \texttt{supported}_{\delta^*} & \wedge & \texttt{beaten}_{\delta^*} \subseteq \texttt{beaten}_{\partial^*} & \wedge \\
\texttt{supported}_{\partial} \subseteq \texttt{supported}_{\delta} & \wedge & \texttt{beaten}_{\delta} \subseteq \texttt{beaten}_{\partial} & \wedge \\
\texttt{defeasibly}_{\partial^*} \subseteq \texttt{supported}_{\partial^*} & \wedge & \texttt{beaten}_{\partial^*} \subseteq \texttt{overruled}_{\partial^*} & \wedge \\
\texttt{defeasibly}_{\delta^*} \subseteq \texttt{defeasibly}_{\partial^*} & \wedge & \texttt{overruled}_{\partial^*} \subseteq \texttt{overruled}_{\delta^*} & \wedge \\
\texttt{defeasibly}_{\partial^*} \subseteq \texttt{supported}_{\delta^*} & \wedge & \texttt{beaten}_{\delta^*} \subseteq \texttt{overruled}_{\partial^*} &
\end{array}
$$

Clearly this statement holds in the empty interpretation. It is mostly straightforward to show that if the induction hypothesis holds in $\Phi \uparrow n$ then it holds in $\Phi \uparrow n{+}1$. For example, consider the first two containments in the induction hypothesis. If they hold in $\Phi \uparrow n$ (and also $\texttt{defeasibly}_{\delta^*} \subseteq \texttt{defeasibly}_{\partial^*}$ holds) then, applying clause $c15$, the second containment holds in $\Phi \uparrow n{+}1$ and, applying clause $c14$, the first containment holds in $\Phi \uparrow n{+}1$. To address fail-expressions we also need the corresponding versions of these containments and arguments for negative conclusions.

One containment, $\texttt{defeasibly}_{\partial} \subseteq \texttt{supported}_{\partial}$ is not easily proved by induction, but it has a direct proof. For a set $S = \Phi \uparrow n$, if $\texttt{defeasibly}_{\partial}(x) \in \Phi(S)$ then there is a supportive rule r whose body literals are defeasibly true in S (i.e. $\texttt{defeasibly}_{\partial}(w_i) \in S$) and $\texttt{not overruled}_{\partial}(r, x) \in S$. Now $\texttt{not overruled}_{\partial}(r, x) \in S$ only if, for every rule s for $\sim x$ whose body literals are defeasibly true in S, there is a supportive rule t whose body literals are defeasibly true in S and $t > s$. Since D is finite and $>$ is acyclic, for some such t, for every such s, $s \not> t$. This t can now be used as r in clauses $c14$ and $c15$ to show that $\texttt{supported}_{\partial}(x) \in \Phi(S)$. □

As mentioned in the body of the paper, it is straightforward to compute the consequences of an annotated defeasible theory in quadratic time. We outline the proof.

Proposition 4
Let D be an annotated defeasible theory, and $|D|$ be the number of symbols in D. Then the set of consequences $\mathcal{C}(D)$ can be computed in time O($|D|^2$).

Proof
(Sketch) Consider the grounding of the clauses, by unfolding with the input representation of the defeasible theory and related facts, and the worst-case (i.e. maximum) size of the result. Unfolding with facts like `proof_tag` produces an increase in rules by a constant factor, because the number of tags is fixed. Unfolding clauses for `rule` etc. produces a set of ground instances linear in the size of rules in D. For clauses $c1$ and $c2$, the size of ground instances is proportional to the size of facts/strict rules in D. The size of ground instances of clauses $c6$, $c14$, and $c15$ is proportional to the size of rules in D. The size of ground instances of clauses $c10$ is proportional to the number of superiority statements in D. The size of ground instances of clauses $c3$ – $c5$ and $c11$ – $c13$ is proportional to the number of literals in D.

For clauses $c7$ and $c8$, the size of the ground instances is proportional to the product of the number of rules in D and the maximum size of rules in D. The size of ground instances of clauses $c9$ is proportional to the product of the number of superiority statements and the maximum size of rules in D.

Thus the size of all ground clauses is bounded above by $|D|^2$. The ground rules form an essentially propositional logic program. Computing the consequences of a propositional logic program under the Kunen semantics is linear in the size of the program. Consequently, the cost of computing the conclusions is O($|D|^2$). □

Recall that $\mathcal{S} = \{partial\ stable, well\text{–}founded, regular, L\text{–}stable, Kunen, Fitting\}$ is a set of semantics. These semantics (and the stable semantics) are preserved by unfolding (with the Kunen semantics requiring the restriction on a rule unfolding itself). This was established for the well-founded (Seki 1993; Aravindan and Dung 1995) and stable models (Maher 1990; Aravindan and Dung 1995), and in (Maher 2017) for the partial stable models and the L-stable models. For the Kunen and Fitting semantics it follows the same proof as in (Maher 1988) for the 2-valued Clark completion semantics. Consequently, Theorem 1 extends to the semantics in $\mathcal{S}$:

Theorem 6
Let $D = (F, R, >)$ be a defeasible theory, and α be an annotation for that theory. Let $d \in \{\delta^*, \delta, \partial^*, \partial\}$. Suppose $\alpha(R)$ contains only annotations `free` and d, and there is no fail-expression in R. Let $S \in \mathcal{S}$. Then

- $\mathcal{M}(\alpha(D)) \models_S \texttt{defeasibly}_{\texttt{d}}(q)$ iff $\mathcal{M}_d(D) \models_S \texttt{defeasibly}(q)$

- $\mathcal{M}(\alpha(D)) \models_S \neg\texttt{defeasibly}_{\texttt{d}}(q)$ iff $\mathcal{M}_d(D) \models_S \neg\texttt{defeasibly}(q)$

and, if $d \in \{\delta^*, \delta\}$,

- $\mathcal{M}(\alpha(D)) \models_S \texttt{supported}_{\texttt{d}}(q)$ iff $\mathcal{M}_d(D) \models_S \texttt{supported}(q)$
- $\mathcal{M}(\alpha(D)) \models_S \neg\texttt{supported}_{\texttt{d}}(q)$ iff $\mathcal{M}_d(D) \models_S \neg\texttt{supported}(q)$

More generally, the S-models of $\mathcal{M}(\alpha(D))$ restricted to $\texttt{defeasibly}_{\texttt{d}}$ are identical (up to predicate renaming) to the S-models of $\mathcal{M}_d(D)$ restricted to $\texttt{defeasibly}$.

Proof
The proof of Theorem 1 also applies to this theorem, since unfolding (without self-unfolding) preserves models for all semantics in $\mathcal{S}$ (see Theorem 3.2 of (Maher 2017)), as does deletion of irrelevant clauses. □

Examples

We present some counterexamples, to show that Figure 1 does not omit any containments and that all the containments are strict. For these examples we do not need to use any annotations: they equally apply to (unannotated) defeasible theories, and we present them in that form.

There are four possible containments we must show do not hold: $\delta \not\subseteq \partial^*$, $\sigma_\delta \not\subseteq \sigma_{\partial^*}$, $\partial \not\subseteq \sigma_\delta$, and $\delta \not\subseteq \sigma_{\partial^*}$. We have two examples that demonstrate these four points.

Example 7
Let the defeasible theory D consist of the rules

$$\begin{array}{rrcr} r_1: & & \Rightarrow & p \\ r_2: & & \Rightarrow & \neg p \\ r_3: & & \Rightarrow & q \\ r_4: & \neg p & \Rightarrow & \neg q \\ r_5: & q & \Rightarrow & s \\ r_6: & & \Rightarrow & \neg s \end{array}$$

with $r_5 > r_6$.

Rules r_1 - r_4 are a standard example distinguishing ambiguity blocking and propagating behaviours. $+\partial^* q$ and $-\delta q$ can be concluded. Thus, $\delta \not\subseteq \partial^*$. In addition, we conclude $-\sigma_{\partial^*}\neg s$ and $+\sigma_\delta \neg s$. Thus, $\sigma_\delta \not\subseteq \sigma_{\partial^*}$.

Now we show that $\partial \not\subseteq \sigma_\delta$ and $\delta \not\subseteq \sigma_{\partial^*}$.

Example 8
Consider the following defeasible theory D:

$$\begin{array}{rrcrcrrcr} r_1: & p & \Rightarrow & \neg q & \qquad & r_5: & & \Rightarrow & p \\ r_2: & & \Rightarrow & q & & r_6: & & \Rightarrow & p \\ r_3: & q & \Rightarrow & \neg s & & r_7: & & \Rightarrow & \neg p \\ r_4: & & \Rightarrow & s & & r_8: & & \Rightarrow & \neg p \end{array}$$

with $r_1 > r_2, r_5 > r_7, r_6 > r_8$

Then we have $+\delta p$ and $-\partial^* p$. Consequently, we have $-\sigma_\delta q$ and $+\partial^* q$. Hence, $\partial \not\subseteq \sigma_\delta$. Furthermore, we have $+\delta s$ and $-\partial^* s$. Hence $\delta \not\subseteq \sigma_{\partial^*}$.

Hence, there are no containments missing from Figure 1.

That the containments in the top row of Figure 1 are strict was mostly established in (Billington et al. 2010). The strictness of containments between forms of support follows straightforwardly from the strictness of containment for the corresponding forms of defeasibility. For the remaining containments, consider the following example.

Example 9
Consider the following defeasible theory D:

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

We have $-\partial^* p$ (and $-\partial^* \neg p$) but $+\sigma_{\delta^*} p$. Consequently, we have $+\partial^* \neg q$ but $-\delta^* \neg q$, showing that $\delta^* \subset \partial^*$ on D.

Note also that we have conclusions $-dp$ and $+\sigma_d p$ for any defeasible proof tag d. Hence $+\partial \subset \sigma_\partial$ and $+\partial^* \subset \sigma_{\partial^*}$

Hence all the containments in Figure 1 are strict.

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.

ARAVINDAN, C. AND DUNG, P. M. 1995. On the correctness of unfold/fold transformation of normal and extended logic programs. *J. Log. Program. 24,* 3, 201–217.

BILLINGTON, D., ANTONIOU, G., GOVERNATORI, G., AND MAHER, M. J. 2010. An inclusion theorem for defeasible logics. *ACM Trans. Comput. Log. 12,* 1, 6.

FITTING, M. 1985. A Kripke-Kleene semantics for logic programs. *J. Log. Program. 2,* 4, 295–312.

MAHER, M. J. 1988. Correctness of a logic program transformation system. Tech. Rep. RC 13496, IBM T.J. Watson Research Center.

MAHER, M. J. 1990. Reasoning about stable models (and other unstable semantics). Tech. rep.

MAHER, M. J. 2017. Relating concrete defeasible reasoning formalisms and abstract argumentation. *Fundam. Inform. 153,* 1, 1–28.

MAHER, M. J. AND GOVERNATORI, G. 1999. A semantic decomposition of defeasible logics. In *AAAI/IAAI*. AAAI Press, 299–305.

SEKI, H. 1993. Unfold/fold transformation of general logic programs for the well-founded semantics. *J. Log. Program. 16,* 1, 5–23.