On Automated Defeasible Reasoning with Controlled Natural Language and Argumentation

Hannes Strass
Computer Science Institute
Leizig University, Germany
strass@informatik.uni-leipzig.de

Adam Wyner
Department of Computing Science
University of Aberdeen, United Kingdom
azwyner@abdn.ac.uk

Abstract

We present an approach to reasoning with strict and defeasible rules over literals. A controlled natural language is employed as human/machine interface to facilitate the specification of knowledge and verbalization of results. Reasoning on the rules is done by a direct semantics that addresses several issues for current approaches to argumentation-based defeasible reasoning. Techniques from formal argumentation theory are employed to justify conclusions of the approach; therefore, we not only address automated reasoning but also human acceptance of provided conclusions.

Introduction

Approaches to artificial intelligence in general and to automated problem solving in particular should be – in virtue of their intelligence – able to explain and justify their conclusions and actions in a rational discourse. This is not always done: the Go playing computer program *AlphaGo* (Silver et al. 2016), while very proficient in choosing the right move (i.e. solving a range of problems), cannot explain to a human user *why* it chose that particular move (i.e. justifying its solution). A recent Nature editorial concluded that "[t]he machine becomes an oracle; its pronouncements have to be believed." (Nature 529, 2016, p. 437)

To make believable, useful results, they have to be communicated to human users, which implies that the formal knowledge models and efficient inference mechanisms ought to be in a familiar, relevant form for humans. In this paper, we aim at addressing specific problems of usability of automated reasoning in a particular, restricted setting. The restricted setting is that of reasoning with non-monotonic semantics of knowledge bases that are given in the form of strict and defeasible rules, since people reason non-monotonically about many matters. For this, we make use of several techniques. Firstly, to address the communication issue (between humans and machines), we employ a *controlled natural language* as specification language for the input of the model as well as the output of inferences. Controlled natural languages (CNLs) are subsets of natural language that have been restricted in lexicon and grammar, thereby eliminating ambiguity and reducing complexity (Kuhn 2014). Some systems automatically translate sentences into formal, machine-readable semantic representations; we adapt one such system, AceRules (Kuhn 2007), for user specification of defeasible theories. Secondly, to address the explanation issue (justifying answers) we employ techniques from formal argumentation theory. Argumentation studies how arguments, which consist of prerequisites, a claim, and an inference between the two, along with their relationships with other arguments, such as rebuttal, determine which arguments are acceptable, that is, which arguments can be defended in a rational discourse. Formal argumentation theory and its implementations formally and automatically construct conclusions from a knowledge base. The CNL interface allows a user to build the knowledge base and to receive justified conclusions in language. We discuss our approach to CNLs and argumentation theory further below.

In contrast to previous approaches that deal with strict and defeasible rules in argumentation theory, in our approach "argument" objects are no longer directly computed upon, but rather constructed as optional by-products for explanation and justification. We show that this novel view addresses a range of problematic issues in existing approaches that are based on Dung's argumentation frameworks (AFs) (1995). These issues are outlined below.

In Dungian AFs, "arguments" are nodes and "attacks" are arcs that indicate some incompatibility between arguments; semantics such as *grounded* or *stable* are provided to calculate sets of arguments that can be interpreted as being collectively acceptable. Existing approaches to give substance to strict and defeasible rules all fall into the realm of *instantiated* abstract argumentation (Besnard and Hunter 2009; Prakken 2010; Bondarenko et al. 1997) (LB, ASPIC+, and ABA, respectively). In such approaches, a knowledge base of strict and defeasible rules over literals is construed as complex "argument" objects (reasoning from prerequisites and rules to conclusions) in attack relations (i.e. contrastiveness between propositions) and then evaluated in a Dungian abstract AF to derive knowledge base conclusions. They do not make use of natural language interfaces.

Such theories must address a range of issues: the rationality postulates (Caminada and Amgoud 2007; Amgoud and Besnard 2013),[1] arguments with subarguments, exponential

[1] The rationality postulates of Caminada and Amgoud essen-

overgeneration of arguments, opacity of attacks, regeneration of arguments when the knowledge base changes, and partial knowledge bases. Moreover, the approaches (except (Besnard and Hunter 2005)) treat propositional knowledge bases while (at least some) elements of predicate logic are needed for any natural language interface to argumentation.

An approach that addresses some of these matters is the work by Wyner et al. (2015), but their approach – despite making use of additional meta-level definitions on top of AF semantics – does not satisfy the rationality postulates (it would violate closure in Example 3 of this paper). Another approach is by Strass (2015), who defines a semantics for defeasible theories based on abstract dialectical frameworks (Brewka and Woltran 2010) and also several direct semantics, but the definitions do not assume that the world is "as normal as possible" (Brewka, Niemelä, and Truszczynski 2008), which is a cornerstone of defeasible reasoning. Moreover, neither of those approaches treats any first-order aspects or connects to a natural language interface.

More generally, the approaches to instantiated argumentation do not strongly tie-in to intuitions about natural language as well as its use. Argument mining (Lippi and Torroni 2016) is promising, but requires extensive (and currently infeasible) preprocessing and normalisation to support formal inference.

In view of the communication aspect, there are controlled natural language tools which translate natural language into First-order Logic expressions and interface to non-monotonic inference engines (Kuhn 2007; Fuchs, Kaljurand, and Kuhn 2008; Fuchs 2016; Guy and Schwitter 2016). Yet, these are not coupled to argumentation or related inference engines. More pointedly, defeasible rules are modeled using 'not provably not', which we show has a different interpretation than the natural expression 'usually' as a normative quantifier over contexts (Kratzer 2012). The following running example is paraphrased from Pollock (2007).

Example 1 (Moustache Murder). *Jones is a person. Paul is a person. Jacob is a person. Usually, a person is reliable. If Jones is reliable then the gunman has a moustache. If Paul is reliable then Jones is not reliable. If Jacob is reliable then Jones is reliable.*

Clearly not both Paul and Jacob can be reliable, and any semantics should be able to "choose" between the two options.

In the approaches of (Fuchs 2016) and (Guy and Schwitter 2016), the adverb of quantification 'usually' is translated as "not provably not" (perhaps along with an abnormality predicate), e.g. a paraphrase for "usually, a person is reliable" is along the lines of "if a person is not provably not reliable then the person is reliable". However, this formalisation can be incorrect,as demonstrated by its straightforward ASP implementation:

```
1: person(jones). person(paul). person(jacob).
2: has(gunman,moustache) :- reliable(jones).
3: -reliable(jones) :- reliable(paul).
4: reliable(jones) :- reliable(jacob).
5: reliable(X) :- person(X), not -reliable(X).
```

This answer set program is inconsistent: Roughly, the literal `-reliable(jacob)` cannot ever be derived from the program, so `reliable(jacob)` must be in every answer set by `(5)` and `(1)`. Thus `reliable(jones)` must be in every answer set by `(4)`. However, the same holds for `paul`, whence the literal `reliable(paul)` must be in every answer set. Thus `-reliable(jones)` must be in every answer set by `(3)`. Consequently, any answer set would have to contain both `reliable(jones)` and `-reliable(jones)`, therefore no answer set exists.[2] Yet, a program ought to produce the intended interpretations as stable models. Thus, the "not provably not" reading of "usually, $\langle conditional \rangle$" phrases is not always correct.[3] In contrast, our approach gets the correct reading as "usually, $\langle conditional \rangle$" is interpreted as a defeasible proposition that holds in as many worlds as consistently possible. ▲

Overgeneration There is a fundamental flaw with using "arguments" as explicit objects to be computed as *there might be just too many of them.* For example, consider the "argument" definition of Caminada and Amgoud (2007) and Prakken (2010), and observe what explicitly creating "argument" objects can amount to computationally:

Example 2. The sequence $(D_n)_{n\in\mathbb{N}}$ of rule sets is given by $D_0 = \{\Rightarrow p_0, \Rightarrow q_0\}$, $D_1 = D_0 \cup \{p_0 \to p_1, q_0 \to p_1\}$ and for all $i \geq 1$, $D_{i+1} = D_i \cup \{p_0, p_i \to p_{i+1}, q_0, p_i \to p_{i+1}\}$. For any $n \in \mathbb{N}$, the size of D_n is linear in n, but D_n leads to 2^{n+1} "arguments", among them 2^n "arguments" for p_n. Here are the sets A_i of "arguments" for D_i for $0 \leq i \leq 2$:

$$\begin{aligned}
A_0 &= \{[\Rightarrow p_0], [\Rightarrow q_0]\}\\
A_1 &= A_0 \cup \{[[\Rightarrow p_0] \to p_1], [[\Rightarrow q_0] \to p_1]\}\\
A_2 &= A_1 \cup \{[[\Rightarrow p_0], [[\Rightarrow p_0] \to p_1] \to p_2],\\
&\qquad [[\Rightarrow p_0], [[\Rightarrow q_0] \to p_1] \to p_2],\\
&\qquad [[\Rightarrow q_0], [[\Rightarrow p_0] \to p_1] \to p_2],\\
&\qquad [[\Rightarrow q_0], [[\Rightarrow q_0] \to p_1] \to p_2]\}
\end{aligned}$$ ▲

The same exponential overgeneration can be observed in assumption-based argumentation (Bondarenko et al. 1997), which uses tree-shaped arguments, and the approach of Amgoud and Nouioua (2015) who essentially use the ASPIC "argument" construction.[4] In recent work, Craven and Toni (2016) addressed some of the computational problems of tree-shaped arguments in ABA; however, their definition of "argument graph" still allows for exactly the above (exponentially many distinct) structures. For the work of Craven and Toni (2016) this is not a substantial problem since they primarily focus on reasoning problems concerned with credulous and sceptical acceptance of conclusions.

tially say that each mutually acceptable set of conclusions as per the semantics must be consistent and closed under strict rules.

[2] We are not claiming that ASP is not adept at treating this example right; we claim that the straightforward "not provably not" reading of "usually, $\langle conditional \rangle$" phrases is not always correct.

[3] Adding an abnormality atom into the body of line 5 (like in rule (12) of (Baral and Gelfond 1994)) would solve the technical problem of inconsistency, but still not get us the intuitive reading we want, and would introduce the problem of having to create abnormality predicates from language input that does not use them.

[4] The D_i are strictly speaking not valid ABA input but can be turned into one using the translation given later in this paper.

However, overgeneration *is* a problem in all approaches that create argument objects and attacks in order to instantiate Dung's abstract AFs. In addition, every time the knowledge base changes, the arguments (and extensions) must be recalculated. Of course, some method may be to block or filter overgeneration; however, it is clearly worthwhile to avoid overgeneration in the first instance. On the technical side, we show in this paper that our approach is as expressive as (flat) ABA for stable semantics. As a corollary, this entails that our proposal is as expressive as ASPIC+ without preferences, since Heyninck and Straßer (2016) have recently shown that the latter can be translated into ABA and thus that both frameworks are equally expressive.

Complex Arguments and Opacity of Attacks Example 2 can also be used to illustrate (at a simple level) complex arguments and opacity of attacks. A_2 contains arguments with subarguments, though the subarguments can only be identified by decomposing the superordinate argument, requiring a further analytic step. Relatedly, were the rule set to have $\Rightarrow \neg q_0$, then the contrasting argument at A_0 is attacked; this attack percolates up to abstractly attack arguments at A_2. Thus, attacks proliferate; if we only looked at arguments at the level of A_2, we would not know precisely the nature of the attack. Of course, workarounds may be feasible, but a better theory would not induce the issue in the first instance.

Contributions of this paper In our approach, we provide interpretations for defeasible theories and then construct argument objects as optional by-products for explanation, justification, and querying. This is in contrast with prevailing approaches that first construct argument objects for a knowledge base and then derive interpretations for knowledge bases from argument extensions. Our approach satisfies the rationality postulates and addresses the range of issues outlined above. In addition, we contribute a new interface between natural language and a defeasible knowledge base, which is largely non-existent in other approaches. This is a useful and straightforward approach since reasoning from the knowledge base, argumentation about it and their natural language counterparts must all be tuned to each other. We have an implementation for our reasoner and apply an existing Controlled Natural Language tool that largely provides the requisite translation to the reasoner's format.

Outline In the rest, we define and exemplify the direct semantics for propositional defeasible theories, outline properties of the approach, define defeasible theories with variables, construct higher level argument structures over theories, and finally tie theories to a natural language interface. We close with some discussion and notes on future work.

Propositional Defeasible Theories

For a set $\mathcal{P}$ of atomic propositions, the set $\mathcal{L}_\mathcal{P}$ of its literals is $\mathcal{L}_\mathcal{P} = \mathcal{P} \cup \{\neg p \mid p \in \mathcal{P}\}$. A *rule* over $\mathcal{L}_\mathcal{P}$ is a pair (B, h) where the finite set $B \subseteq \mathcal{L}_\mathcal{P}$ is called the *body* (premises) and the literal $h \in \mathcal{L}_\mathcal{P}$ is called the *head* (conclusion). For $B = \{b_1, \ldots, b_k\}$ with $k \in \mathbb{N}$, we sometimes write rules in a different way: a *strict* rule is of the form "$b_1, \ldots, b_k \rightarrow h$"; a *defeasible* rule is of the form "$b_1, \ldots, b_k \Rightarrow h$". In case $k = 0$ we call "$\rightarrow h$" a *fact* and "$\Rightarrow h$" an *assumption*.

The intuitive meaning of a rule (B, h) is that whenever we are in a state of affairs where all literals in B hold, then also literal h holds. Given a set L of literals representing a state of the world, a rule (B, h) is *applicable* if $B \subseteq L$ and *inapplicable* otherwise. We say that a rule (B, h) *holds* for a set L of literals if $B \subseteq L$ implies $h \in L$. (Put another way, (B, h) holds for L iff $B \cup \{h\} \subseteq L$ or $B \nsubseteq L$.) So a rule makes a statement about a world, and can hold for one world but possibly not so for another. For example, the rule $(\{a\}, b)$ holds in the worlds $\{a, b\}$ and $\emptyset$ but not in $\{a\}$ or $\{a, \neg b\}$. In particular, a rule $(\{a\}, b)$ is not equivalent to its contrapositive $(\{\neg b\}, \neg a)$, as the former holds in the world $\{\neg b\}$ but the latter does not. Thus rules are not to be confused with material implication in propositional logic.

The difference between strict and defeasible rules is the following: A strict rule *must* hold in *all* possible worlds. A defeasible rule *should* hold in *most* possible worlds. That is, there might be some worlds that are exceptional with respect to some defeasible rules, but we can still consider those worlds possible. On the other hand, a world where some strict rule does not hold is impossible.

A *defeasible theory* is a tuple $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ where $\mathcal{P}$ is a set of atomic propositions, $\mathcal{S}$ is a set of strict rules over $\mathcal{L}_\mathcal{P}$ and $\mathcal{D}$ is a set of defeasible rules over $\mathcal{L}_\mathcal{P}$. The meaning of defeasible theories is defined as follows. To define the meta-level negation of literals, we define $\overline{p} = \neg p$ and $\overline{\neg p} = p$ for $p \in P$. A set $L \subseteq \mathcal{L}_\mathcal{P}$ of literals is *consistent* iff for all $z \in \mathcal{L}_\mathcal{P}$ we find that $z \in L$ implies $\overline{z} \notin L$. For a set $R \subseteq \mathcal{S} \cup \mathcal{D}$ of rules and a set $L \subseteq \mathcal{L}_\mathcal{P}$ of literals, we define $R(L) = \{h \in \mathcal{L}_\mathcal{P} \mid (B, h) \in R, B \subseteq L\}$; a set L of literals is *closed under* R iff $R(L) \subseteq L$. We next present the first bit of our direct semantics. The main underlying intuition goes back to foundational work on the treatment of inconsistency by Rescher and Manor (1970), and to work on defeasible logical reasoning by Poole (1988).

Definition 1. Let $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ be a defeasible theory. A set $M \subseteq \mathcal{L}_\mathcal{P}$ of literals is a *possible set* for $\mathcal{T}$ if and only if there exists a set $\mathcal{D}_M \subseteq \mathcal{D}$ such that:

1. M is consistent;
2. M is closed under $\mathcal{S} \cup \mathcal{D}_M$;
3. $\mathcal{D}_M$ is $\subseteq$-maximal with respect to items 1 and 2. ▲

Intuitively, a possible set of literals is consistent, closed under strict rules and maximally consistent with respect to applying defeasible rules. It follows that each possible set M induces a set $\mathcal{D}_M$ of defeasible rules that hold in M.

Not every defeasible theory has possible sets:

Example 3. The theory $(\{a\}, \{\rightarrow a, \quad a \rightarrow \neg a\}, \emptyset)$ does not have a possible set, as in any candidate L we have $a \in L$ and by closure also $\neg a \in L$, thus violating consistency. ▲

Regarding the "usually, if B then h" reading of a defeasible rule (B, h), the maximality condition in Definition 1 ensures that possible sets are as "usual" as possible (with respect to the given rules). But in a possible set, there might still be cyclic or otherwise unjustified conclusions.

Example 4. Consider $\mathcal{T} = (\{a, b\}, \emptyset, \{a \Rightarrow b, \quad b \Rightarrow a\})$, a simple defeasible theory with seven possible sets, $M_1 = \emptyset$, $M_2 = \{\neg a\}$, $M_3 = \{\neg b\}$, $M_4 = \{\neg a, \neg b\}$, $M_5 = \{a, \neg b\}$,

$M_6 = \{\neg a, b\}$, $M_7 = \{a, b\}$. Almost all of the possible sets (except $M_1 = \emptyset$) contain unjustified conclusions. For example in M_2, literal $\neg a$ is just there although there is no rule support for it. Likewise, in M_7, literal a holds because b does and vice versa. In some contexts, e.g. causal reasoning (Denecker, Theseider-Dupré, and van Belleghem 1998), a model like M_7 can be unintended as there is no "outside" support (no causal reasons) for any of a, b. ▲

Below, we further refine our direct semantics to rule out interpretations where some literals cannot be justified. We start with the notion of a derivation, which is basically a proof of a literal using only modus ponens over rules.

Definition 2. Let $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ be a defeasible theory. A *derivation in* $\mathcal{T}$ is a set $R \subseteq \mathcal{S} \cup \mathcal{D}$ of rules with a partial order $\preccurlyeq$ on R such that:

1. $\preccurlyeq$ has a greatest element $(B_z, z) \in R$;
2. for each rule $(B, h) \in R$, we have: for each $y \in B$, there is a rule $(B_y, y) \in R$ with $(B_y, y) \prec (B, h)$ (where $\prec$ is the strict partial order contained in $\preccurlyeq$);
3. R is $\subseteq$-minimal with respect to items 1 and 2. ▲

Intuitively, a derivation always concludes some specific unique literal z via a rule (B_z, z), and then in turn contains derivations for all $y \in B_z$ needed to derive z, and so on, down to facts and assumptions. Minimality ensures that there are no spurious rules that are not actually needed to derive z. The partial order $\preccurlyeq$ guarantees that derivations are acyclic. For the above, we say that R *is a derivation for* z.

Example 5. Consider the defeasible theory $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ with $\mathcal{P} = \{a, b, c\}$, strict rules $\mathcal{S} = \{\rightarrow a, \quad a, b \rightarrow c\}$, and defeasible rules $\mathcal{D} = \{\Rightarrow b, \quad a \Rightarrow c\}$. There are two distinct derivations for the literal c, where the order of presentation reflects the ordering $\prec$ on the rules:

$$d_1 = \{(\emptyset, a), (\emptyset, b), (\{a, b\}, c)\} \mathrel{\hat{=}} \{\rightarrow a, \Rightarrow b, \quad a, b \rightarrow c\}$$
$$d_2 = \{(\emptyset, a), (\{a\}, c)\} \mathrel{\hat{=}} \{\rightarrow a, \quad a \Rightarrow c\}$$ ▲

Now we refine the direct semantics such that only literal sets with derivations for all its elements are considered.

Definition 3. Let $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ be a defeasible theory and $M \subseteq \mathcal{L}_{\mathcal{P}}$ be a possible set for $\mathcal{T}$. M is a *stable set for* $\mathcal{T}$ iff for every $z \in M$ there is a derivation of z in $(\mathcal{P}, \mathcal{S}, \mathcal{D}_M)$. ▲

Thus stable sets are possible sets where all contained literals are grounded in facts and assumptions. It does not matter which of the two – there is no ontological distinction between defeasible and strict rules on the level of a single stable set. Intuitively, a stable set is a coherent, *justified* set of beliefs in which the world is as normal as possible. Each stable set M is uniquely characterized by a set $\mathcal{D}_M$ of applied defeasible rules; we will sometimes make use of this herein.

Properties of the Direct Semantics

Rationality Postulates It is immediate from Definition 1 (possible sets) that the semantics satisfies the rationality postulates closure and direct consistency (Caminada and Amgoud 2007), simply because they are built into the definition.

Proposition 1. *Let $\mathcal{T}$ be a defeasible theory. All possible sets of $\mathcal{T}$ are consistent and closed under strict rules.*

The satisfaction of indirect consistency, and the same properties for stable sets then follow as easy corollaries.

Formal Expressiveness With regard to the measure of being able to express sets of two-valued interpretations (Gogic et al. 1995), is quite clear that our approach is as expressive as propositional logic. Consider a propositional formula φ over a propositional vocabulary P. φ can be transformed into an equivalent formula ψ in conjunctive normal form, that is, of the form $\psi = \psi_1 \land \ldots \land \psi_n$ where each ψ_i is a disjunction of literals.[5] We create a defeasible theory $\mathcal{T}_\varphi = (P, \mathcal{S}_\varphi, \mathcal{D}_P)$ as follows: the defeasible rules are $\mathcal{D}_P = \{\Rightarrow p, \Rightarrow \neg p \mid p \in P\}$; for each conjunct $\psi_i = \psi_i^1 \lor \ldots \lor \psi_i^{m_i}$ of ψ, the set $\mathcal{S}_\varphi$ contains the strict rules $\overline{\psi_i^2}, \overline{\psi_i^3}, \ldots, \overline{\psi_i^{m_i}} \rightarrow \psi_i^1$, $\overline{\psi_i^1}, \overline{\psi_i^3}, \ldots, \overline{\psi_i^{m_i}} \rightarrow \psi_i^2$, $\ldots$, $\overline{\psi_i^1}, \overline{\psi_i^2}, \ldots, \overline{\psi_i^{m_i - 1}} \rightarrow \psi_i^{m_i}$. (Intuitively, these rules correspond to all transpositions of the disjunction ψ_i.)

Proposition 2. *For any propositional formula φ, the stable sets of $\mathcal{T}_\varphi$ correspond one-to-one with the models of φ.*

Relationship to ABA Our approach can faithfully model flat ABA for stable semantics. Let $(\mathfrak{L}, \mathfrak{R}, \mathfrak{A}, \hat{\cdot})$ be a flat ABA framework (Bondarenko et al. 1997; Toni 2014). We define the defeasible theory $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ with vocabulary $\mathcal{P} = \mathfrak{L}$, strict rules $\mathcal{S} = \mathfrak{R} \cup \{\hat{a} \rightarrow \neg a \mid a \in \mathfrak{A}\}$, and defeasible rules $\mathcal{D} = \{\Rightarrow a \mid a \in \mathfrak{A}\}$. (For the purposes of this translation, we treat the elements of $\mathfrak{L}$ as atomic entities.) Intuitively, the elements of the original language underlying the given ABA framework $\mathfrak{F}$ are considered the atoms of the resulting defeasible theory language. The strict rules of the original language persist, and are enriched by additional rules $\hat{a} \rightarrow \neg a$ that encode the meaning of contraries via classical negation. The created defeasible rules implement the intended meaning of assumptions, namely that they can be assumed without justification.

Theorem 3. *Let $\mathfrak{F} = (\mathfrak{L}, \mathfrak{R}, \mathfrak{A}, \hat{\cdot})$ be an ABA framework and $\mathcal{T}$ be its corresponding defeasible theory according to the above definition. The stable sets of $\mathcal{T}$ correspond one-to-one with the stable sets of assumptions of $\mathfrak{F}$.*

We conjecture that a translation in the converse direction can be done in a similar way: starting from $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$, we define the ABA framework $(\mathfrak{L}, \mathfrak{R}, \mathfrak{A}, \hat{\cdot})$ with language $\mathfrak{L} = \mathcal{L}_{\mathcal{P}} \cup \mathfrak{A}$ where $\mathfrak{A} = \{a_d \mid d \in \mathcal{D}\}$, derivation rules $\mathfrak{R} = \mathcal{S} \cup \left\{(B \cup \left\{a_{(B,h)}\right\}, h) \,\middle|\, (B, h) \in \mathcal{D}\right\}$, and the contrary of an assumption $a_{(B,h)} \in \mathfrak{A}$ is the literal $\overline{h}$.

On the other hand, our approach offers the possible-set semantics, which is beyond ABA in the sense that ABA has groundedness of conclusions built into its core (via argument construction). In Example 4, there are several pairs of possible sets M' and M'' such that $M' \subsetneq M''$. Such a situation is not easily reproducible with ABA (under stable semantics) and would need the introduction of additional technicalities.

Computational Complexity We first analyze the most important decision problems associated with our direct seman-

[5] Since we are only interested in expressiveness here, a potential exponential blowup during CNF conversion is of no interest.

tics, namely stable set verification, stable set existence, and credulous and sceptical reasoning. Unfortunately, there is no space for the (quite technical) proofs, where we show hardness for the first two items via original reductions and do the same for the last two items via a reduction from (the complement of) the second item.

Theorem 4. *1. The problem "given a defeasible theory $\mathcal{T}$ and a set $M \subseteq \mathcal{L}_{\mathcal{P}}$ of literals, decide whether M is a stable set of $\mathcal{T}$" is* coNP*-complete.*

2. The problem "given a defeasible theory $\mathcal{T}$, decide whether it has a stable set" is Σ_2^P-complete.

3. The problem "given a defeasible theory $\mathcal{T}$ and a literal $z \in \mathcal{L}_{\mathcal{P}}$, decide whether z is contained in some stable set of $\mathcal{T}$" is Σ_2^P-complete.

4. The problem "given a defeasible theory $\mathcal{T}$ and a literal $z \in \mathcal{L}_{\mathcal{P}}$, decide whether z is contained in all stable sets of $\mathcal{T}$" is Π_2^P-complete.

Now for computing explanations for conclusions. Once we have obtained a stable set M for a defeasible theory $\mathcal{T}$ and are given a literal $z \in M$ along with the question of why z is true in M, our task is to compute a derivation of z in the theory $(\mathcal{P}, \mathcal{S}, \mathcal{D}_M)$. For that task, we can employ the fact that rules of our defeasible theories can – ignoring negation – be seen as definite Horn clauses. For definite Horn clauses in propositional logic, in turn, it is well-known that computing a proof for a conclusion can be done in polynomial time.

Reasoning by cases By definition, stable-set semantics does not do reasoning by cases, that is, does not explicitly consider that literals might hold for unspecified reasons. Wyner et al. (2015) have argued why and when such behaviour can be useful, for example when dealing with incompletely specified knowledge bases. Our possible-set semantics of Definition 1 naturally does reasoning by cases and still satisfies the rationality postulates; it thus can be seen as combining the strengths of both approaches.

Implementation We implemented our semantics in (disjunctive) answer set programming (Gebser et al. 2012). For representing defeasible theories, rules are identified by ASP terms. The binary predicates `head/2` and `body/2` declare rule heads and bodies, respectively; predicate `def/1` declares a rule to be defeasible. The implemenation consists of a reasonably small encoding of Definition 3 into ASP; the maximization aspects are implemented using saturation techniques. The encoding works such that the union of the encoding together with the specification of a defeasible theory is given to a solver, and the answer sets of the resulting logic program union correspond one-to-one to the stable sets of the defeasible theory. The implementation is available at github: `https://github.com/hstrass/defeasible-rules`.

Defeasible Theories with Variables

Having seen a language for defeasible reasoning and analyzed some of its formal properties, in this section we add a limited set of first-order features that bring this language closer to natural language, as discussed in a later section. The first step, in this section, will add predicates, variables and constants to the language. This will enable us to express properties of and relationships between objects, to make repeated references to objects and it provides a limited form of universal quantification. The resulting language of defeasible rules follows standard logical (Herbrand-style) approaches and will still be essentially propositional (Schulz 2002; Lierler and Lifschitz 2013) and thus still effectively decidable by the same bounds established earlier.

Syntax Let $\mathcal{V} = \{x_0, x_1, x_2, \ldots\}$ be a countable set of first-order variables and $\mathcal{C}$ be a finite set of constants, that is, null-ary function symbols. For a finite first-order predicate signature $\Pi = \{p_1/k_1, \ldots, p_n/k_n\}$ (where p_i/k_i denotes that p_i is a predicate with arity k_i), the set of all atoms over Π, $\mathcal{V}$ and $\mathcal{C}$ is $\mathit{atoms}(\Pi, \mathcal{V} \cup \mathcal{C}) = \{p(t_1, \ldots, t_k) \mid p/k \in \Pi \text{ and } t_1, \ldots, t_k \in \mathcal{V} \cup \mathcal{C}\}$. A *defeasible theory with variables* is of the form $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ where $\mathcal{P} \subseteq \mathit{atoms}(\Pi, \mathcal{V} \cup \mathcal{C})$ and (as usual) $\mathcal{S}$ and $\mathcal{D}$ are sets of (strict and defeasible, respectively) rules over literals $\mathcal{L}_{\mathcal{P}}$. In particular, rules can mention variables.

Semantics The semantics of defeasible theories with variables is defined via ground instantiation. A *ground substitution* is a function $\gamma : \mathcal{V} \to \mathcal{C}$. Applying a ground substitution γ to a rule works via its homomorphic continuation $\tilde{\gamma}$: $\tilde{\gamma}((B, h)) = (\{\tilde{\gamma}(b) \mid b \in B\}, \tilde{\gamma}(h))$, where for $P/n \in \Pi$ we have $\tilde{\gamma}(P(t_1, \ldots, t_n)) = P(\tilde{\gamma}(t_1), \ldots, \tilde{\gamma}(t_n))$ with $\tilde{\gamma}(c) = c$ for all $c \in \mathcal{C}$ and $\tilde{\gamma}(v) = \gamma(v)$ for all $v \in \mathcal{V}$. The grounding of a defeasible theory with variables $\mathcal{T} = (\mathit{atoms}(\Pi, \mathcal{V}, \mathcal{C}), \mathcal{S}, \mathcal{D})$ has a vocabulary of all ground atoms and contains all ground instances of its rules:

$$\mathit{ground}(\mathcal{T}) = (\mathit{atoms}(\Pi, \mathcal{C}), \mathit{ground}(\mathcal{S}), \mathit{ground}(\mathcal{D}))$$
$$\mathit{ground}(\mathcal{R}) = \{\gamma(r) \mid r \in \mathcal{R}, \gamma : \mathcal{V} \to \mathcal{C}\}$$

A set $M \subseteq \mathcal{L}_{\mathit{atoms}(\Pi,\mathcal{C})}$ is a *stable set for a defeasible theory with variables* $\mathcal{T}$ iff M is a stable set of $\mathit{ground}(\mathcal{T})$.

We illustrate the language with our running example.

Example 1 (Continued). The text on the gunman mystery from earlier leads to this defeasible theory with variables:

$$\begin{aligned}
&\Pi = \{\mathit{person}/1, \mathit{reliable}/1, \mathit{has}/2\}\\
&\mathcal{C} = \{\mathit{jones}, \mathit{paul}, \mathit{jacob}, \mathit{gunman}, \mathit{moustache}\}\\
&\mathcal{T} = (\mathit{atoms}(\Pi, \mathcal{V}, \mathcal{C}), \mathcal{S}, \mathcal{D})\\
&\mathcal{S} = \{\to \mathit{person}(\mathit{jones}), \to \mathit{person}(\mathit{paul}), \to \mathit{person}(\mathit{jacob}),\\
&\qquad \mathit{reliable}(\mathit{jones}) \to \mathit{has}(\mathit{gunman}, \mathit{moustache}),\\
&\qquad \mathit{reliable}(\mathit{paul}) \to \neg\mathit{reliable}(\mathit{jones}),\\
&\qquad \mathit{reliable}(\mathit{jacob}) \to \mathit{reliable}(\mathit{jones})\}\\
&\mathcal{D} = \{\mathit{person}(x_1) \Rightarrow \mathit{reliable}(x_1)\}
\end{aligned}$$

This defeasible theory has two stable sets:

$$\begin{aligned}
M_1 &= M \cup \{\mathit{reliable}(\mathit{jacob}), \mathit{reliable}(\mathit{jones}),\\
&\qquad \mathit{has}(\mathit{gunman}, \mathit{moustache})\} \text{ and}\\
M_2 &= M \cup \{\mathit{reliable}(\mathit{paul}), \neg\mathit{reliable}(\mathit{jones})\}, \text{ with}\\
M &= \{\mathit{person}(\mathit{jones}), \mathit{person}(\mathit{paul}), \mathit{person}(\mathit{jacob})\}.
\end{aligned}$$

Thus our stable-set semantics makes a choice whether Jacob is reliable or Paul is reliable, avoiding inconsistency. ▲

Three Senses of "Argument"

Wyner et al. (2015) provided an analysis of the different terminological meanings of the word "argument" and how the term is used in instantiated abstract argumentation. In their view, there are three distinct (although related) meanings of "argument": (i) a one-step reason for a claim (also called *argument* in this paper), (ii) a chain of reasoning leading towards a claim (a *case*), (iii) reasons for and against a claim (a *debate*). Wyner et al. (2015) then went on to define an AF-based approach for dealing with problems that they observed to result from conflating the three senses in existing work. Although technically their approach falls short of satisfying all our needs, we nevertheless agree with their initial analysis. In what follows, we show how the three different senses of "argument" according to Wyner et al. (2015) appear as distinct entities in the approach of this paper.

Definition 4. Let $\mathcal{T} = (\mathcal{P}, \mathcal{S}, \mathcal{D})$ be a defeasible theory, $M \subseteq \mathcal{L}_\mathcal{P}$ be a stable set of $\mathcal{T}$ and $z \in \mathcal{L}_\mathcal{P}$ be a literal.

- An *argument for z from M* is a rule $(B, z) \in \mathcal{S} \cup \mathcal{D}_M$.
- A *case for z in M* is a derivation for z in $(\mathcal{P}, \mathcal{S}, \mathcal{D}_M)$.
- A *debate about z* is a pair $\langle C^+, C^- \rangle$ of sets of cases, where C^+ only contains cases for z and C^- only contains cases for $\overline{z}$, i.e., cases against z. ▲

Intuitively, an argument is just an atomic deduction where a single rule (B, h) of the defeasible theory is used to make the claim "h holds because all of B hold." A case involves a whole chain of reasoning (possibly involving several arguments building on top of one another) that must be grounded in facts and assumptions, and internally consistent (as witnessed by there being a stable set where the derivation applies). A debate, in turn, involves several cases that might originate from different (possibly incompatible) stable sets.

Example 1 (Continued). In Pollock's moustache example, the derivation

$$\begin{aligned} C_1 = \{ &\to \mathit{person}(\mathit{jones}), \\ &\mathit{person}(\mathit{jones}) \Rightarrow \mathit{reliable}(\mathit{jones}), \\ &\mathit{reliable}(\mathit{jones}) \Rightarrow \mathit{has}(\mathit{gunman}, \mathit{moustache}) \} \end{aligned}$$

is a case for $\mathit{has}(\mathit{gunman}, \mathit{moustache})$ in M_1, and so is

$$\begin{aligned} C_2 = \{ &\to \mathit{person}(\mathit{jacob}), \\ &\mathit{person}(\mathit{jacob}) \Rightarrow \mathit{reliable}(\mathit{jacob}), \\ &\mathit{reliable}(\mathit{jacob}) \to \mathit{reliable}(\mathit{jones}), \\ &\mathit{reliable}(\mathit{jones}) \Rightarrow \mathit{has}(\mathit{gunman}, \mathit{moustache}) \} \end{aligned}$$

Both of these cases contain (sub-)derivations that are cases for $\mathit{reliable}(\mathit{jones})$. We can also construct a case for the opposite literal $\neg\mathit{reliable}(\mathit{jones})$ in M_2:

$$\begin{aligned} C_3 = \{ &\to \mathit{person}(\mathit{paul}), \\ &\mathit{person}(\mathit{paul}) \Rightarrow \mathit{reliable}(\mathit{paul}), \\ &\mathit{reliable}(\mathit{paul}) \to \neg\mathit{reliable}(\mathit{jones}) \} \end{aligned}$$

Taking the sub-cases of C_1 and C_2 for $\mathit{reliable}(\mathit{jones})$ and C_3 together leads to a debate about $\mathit{reliable}(\mathit{jones})$. ▲

For the rule set $\mathcal{S} = \{\to p, \quad p \to p\}$, which is troublesome for approaches with nested "arguments", our definitions above just yields two arguments for p, of which only one ($\to p$) leads to a case for p. For Example 2, our definition would also lead to an exponential number of derivations for each p_n; the important difference to previous approaches is that *we do not explicitly compute on them*. Derivations only become relevant *after* the semantics is computed.

The senses of argument here are related to, but different from, arguments in AF analyses of instantiated argumentation. An argument in Definition 4 is just a rule in ASPIC+, LBA, or ABA, where arguments require a deduction. A case in Definition 4 is an argument in these other approaches. A debate in Definition 4 is a Rebuttal attack in ASPIC+ and LBA. We have no undercutters since as of yet rules have no names; however, notionally an undercutter has applied when a defeasible rule does not appear in an extension.

Obtaining Defeasible Theories from Controlled Natural Language

In this section, we argue for using a Controlled Natural Language (CNL) as an interface to argumentation using direct semantics, whereby natural language input is subjected to automatic analysis (parsing, semantic representation), then reasoned with (direct semantics), and finally output in natural language. Our proposal is the first to facilitate automatic reasoning from inconsistent knowledge bases in natural language (Kuhn 2007; Fuchs, Kaljurand, and Kuhn 2008; Guy and Schwitter 2016). We touch on the main themes.

Argumentation and natural language processing has been an area of intense, recent research (Lippi and Torroni 2016). In argument mining, texts are extracted from unstructured natural language corpora, then mapped to arguments for reasoning in Dungian AFs. Machine learning techniques are applied to identify topics, classify statements as claim or justification, or relate contrasting statements. However, natural language is highly complex and diverse in lexicon, syntax, semantics, and pragmatics. Current mining approaches do not systematically address matters of synonymy, contradiction, or deductions, which require *fine-grained* analysis into a formal language such as Predicate Logic (also see the *recognizing textual entailment* tasks (Androutsopoulos and Malakasiotis 2010)).

We take a different approach, instead working with a *controlled* natural language (CNL) (Kuhn 2014), which restricts the lexicon and grammar as well as disambiguates sentences. More specifically, we work with Attempto Controlled English (ACE) (Fuchs, Kaljurand, and Kuhn 2008; Kuhn 2007) (also see RACE (Fuchs, Kaljurand, and Kuhn 2008) and PENG-ASP (Guy and Schwitter 2016)). ACE translates the input language to machine-readable, First-order Logic expressions and interfaces with inference engines for model generation and theorem proving. ACE facilitates an *engineered solution* to argumentation in NL by addressing three critical issues. It provides *normalised language* which, in principle, can serve as target expressions for information extracted by argument mining; thus we can process arguments and reason in the requisite way. We can *experimentally control and augment the language* input as needed. ACE gives us an *essential experimental interface with inference engines*, enabling testing of different forms and combinations of transformations from natural language to a formal language, then the interaction with alternative inference engines. Finally, a formal, engineered approach helps to scope and systematically resolve issues.

We used AceRules (Kuhn 2007), a sublanguage of ACE. We select and briefly comment on AceRules. AceRules has a range of lexical components: proper names, common nouns, logical connectives, existential and universal quantifiers, one and two place predicates, and relative clauses. Construction rules define the admissible sentence structures, e.g. declarative or conditional sentences. Interpretation rules disambiguate admissible sentences and constrain their logical analysis, while discourse representation accounts for pronominal anaphora. There are further lexical elements and syntactic constructions to use as needed. Verbalisation generates natural language expressions from the formal representations. A range of auxiliary axioms (from ACE) can be optionally added to treat generic linguistic inferences, e.g. interpretations of "be", relations between the plural and the singular form of nouns, and lexical semantic inferences such as *throw* implies *move*. Domain knowledge must be added as well into AceRules.

Turning to semantical key issues, AceRules has linguistic expressions for strong negation, negation-as-failure, the strict conditional, and the adverb 'usually' on events. It connects to different inference engines (courteous logic programs, stable models, and stable models with strong negation) and allows others, e.g. our direct semantics.[6] These features are sufficient to reason non-monotonically. However, there are two key problems with AceRules (and shared with RACE and PENG-ASP): it cannot reason from inconsistent knowledge bases (as in the Nixon diamond example), and it does not incorporate the defeasible conditional. We have argued that both are essential for argumentation. We have shown (see Example 1) that a conditional with 'not provably not' is not semantically equivalent to the natural interpretation of 'usually $\langle \textit{conditional} \rangle$' as the defeasible conditional. To address the first problem, AceRules logical forms are evaluated with respect to our direct semantics. For the second problem, we have manually represented 'usually $\langle \textit{conditional} \rangle$' as a defeasible conditional.[7]

As with all CNLs, care must be taken to input statements in AceRules since they must comply with the language conventions. Terminology may need to be introduced. Information that might be presupposed in natural language must be made explicit. Importantly, one must check that the output semantic representations conform to the intended meaning, and where not, create a paraphrase that yields the intended meaning. For Example 1, we have explicitly stated *There is a gunman*, which might be presupposed. Furthermore, "a person" in *Usually, a person is reliable* is generic; to conform to AceRules and our adaption of defeasible rules, this is rendered as *Usually, if someone X is a person then X is reliable.* Otherwise, we can input the example sentences to AceRules, which parses and represents them, processes them through the direct semantics, and verbalises the result as intended. AceRules has been sufficient for inputting several standard examples from the argumentation literature – *Tweety*, *Nixon*, or *Tandem* – and receiving the correct direct semantic outputs. This thereby delivers a proof of concept.

Discussion and Future Work

We introduced and analysed a direct semantics for defeasible theories (with variables), and tied this semantics to input (text) and output (verbalisation) in natural language.

Although we argue for our approach from first principles, several of its elements have precursors in the literature. For one, (Dung and Son 2001) define "defeasible derivations" (which need not be minimal but are otherwise just like our derivations) and a (stable) extension semantics without explicit argument construction (which is similar to the stable set semantics of our approach). Moreover, Amgoud and Besnard (2013) have a notion that is similar to our semantics: for a stable set M, they would call $(\mathcal{P}, \mathcal{S}, \mathcal{D}_M)$ an "option" of $\mathcal{T}$. Finally, our notion of derivation is similar to what Craven and Toni (2016) would call a "focussed, rule-minimal argument graph". In slightly more distant related work, Denecker, Brewka, and Strass (2015) introduced a general theory of justifications, where there are also rules involving literals; they however do not have a natural language interface and a decidedly more philosophical/mathematical motivation. For example, they allow infinite justifications, which is not immediately useful for our setting. A more argumentation-oriented approach is that by Schulz and Toni (2016), who provide tree-shaped justifications for why literals are elements of answer sets of a given logic program, albeit they do not deal with natural language.

Another approach to avoiding inconsistency in ASP formalisations of "usually, Ps are Qs" are *consistency-restoring rules* (Balduccini and Gelfond 2003). Since those rules have to be added to the program, that approach is somewhat orthogonal to ours, where the semantics disregards some of the specified defeasible rules to obtain consistency.

For future work, our approach could be extended to deal with inconsistencies that arise purely among strict rules by identifying minimal inconsistent rule-subsets and "downgrading" the strict rules therein to defeasible rules. An extreme form of this 'downgrading' happens in the approach of Besnard and Hunter (2009), where all elements of a given (possibly inconsistent) knowledge base are considered defeasible (as an analogue of closure need not hold for them).

Another major point of future work is extending the semantics for priorities among defeasible rules, for example by using ideas from preferred subtheories (Brewka 1989). Clearly, the resulting semantics should satisfy the rationality postulates of Dung (2016). A major area of future work is the overall pipeline passing through natural language, formal representation, argument semantics, and verbalisation.

[6] RACE and PENG-ASP have the same expressions (Fuchs, Kaljurand, and Kuhn 2008; Guy and Schwitter 2016). RACE is based on *Satchmo* (written in Prolog), while PENG-ASP uses ASP.

[7] An integration to AceRules is feasible; see, in a related setting, *If Nixon is a quaker then Nixon usually is a pacifist.* in `https://argument-pipeline.herokuapp.com/`, which is based on (Wyner et al. 2015).

References

Amgoud, L., and Besnard, P. 2013. A formal characterization of the outcomes of rule-based argumentation systems. In Liu, W.; Subrahmanian, V. S.; and Wijsen, J., eds., *Scalable Uncertainty Management – 7th International Conference, SUM 2013, Washington, DC, USA, September 16–18, 2013. Proceedings*, volume 8078 of *Lecture Notes in Computer Science*, 78–91. Springer.

Amgoud, L., and Nouioua, F. 2015. Undercutting in argumentation systems. In Beierle, C., and Dekhtyar, A., eds., *Scalable Uncertainty Management – 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings*, volume 9310 of *Lecture Notes in Computer Science*, 267–281. Springer.

Androutsopoulos, I., and Malakasiotis, P. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research* 38:135–187.

anonymous. 2016. Digital intuition: A computer program that can outplay humans in the abstract game of Go will redefine our relationship with machines. *Nature* 529:437. Editorial.

Balduccini, M., and Gelfond, M. 2003. Logic programs with consistency-restoring rules. In *International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2003 Spring Symposium Series*, volume 102.

Baral, C., and Gelfond, M. 1994. Logic programming and knowledge representation. *Journal of Logic Programing* 19/20:73–148.

Besnard, P., and Hunter, A. 2005. Practical first-order argumentation. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, 590–595. AAAI Press.

Besnard, P., and Hunter, A. 2009. Argumentation based on classical logic. In Rahwan, I., and Simari, G., eds., *Argumentation in Artificial Intelligence*. Springer. 133–152.

Bondarenko, A.; Dung, P. M.; Kowalski, R. A.; and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93:63–101.

Brewka, G., and Woltran, S. 2010. Abstract dialectical frameworks. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Represetnation and Reasoning (KR 2010)*, 102–211.

Brewka, G.; Niemelä, I.; and Truszczynski, M. 2008. Preferences and nonmonotonic reasoning. *AI Magazine* 29(4):69–78.

Brewka, G. 1989. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of the Eleventh International Conference on Artificial Intelligence*, 1043–1048.

Caminada, M., and Amgoud, L. 2007. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171(5–6):286–310.

Craven, R., and Toni, F. 2016. Argument graphs and assumption-based argumentation. *Artificial Intelligence* 233:1–59.

Denecker, M.; Brewka, G.; and Strass, H. 2015. A formal theory of justifications. In Calimeri, F.; Ianni, G.; and Truszczyński, M., eds., *Proceedings of the Thirteenth International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR)*, 250–264. Lexington, KY, USA: Springer-Verlag Berlin Heidelberg.

Denecker, M.; Theseider-Dupré, D.; and van Belleghem, K. 1998. An Inductive Definition Approach to Ramifications. *Linköping Electronic Articles in Computer and Information Science* 3(7):1–43.

Dung, P. M., and Son, T. C. 2001. An argument-based approach to reasoning with specificity. *Artificial Intelligence* 133(1-2):35–85.

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.

Dung, P. M. 2016. An axiomatic analysis of structured argumentation with priorities. *Artificial Intelligence* 231:107–150.

Fuchs, N. E.; Kaljurand, K.; and Kuhn, T. 2008. Attempto Controlled English for knowledge representation. In *Reasoning Web*, 104–124.

Fuchs, N. E. 2016. Reasoning in Attempto Controlled English: Non-monotonicity. In Davis, B.; Pace, G. J.; and Wyner, A., eds., *Controlled Natural Language – 5th International Workshop, CNL 2016, Aberdeen, UK, July 25-27, 2016, Proceedings*, volume 9767 of *Lecture Notes in Computer Science*, 13–24. Springer.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2012. Answer set solving in practice. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(3):1–238.

Gogic, G.; Kautz, H.; Papadimitriou, C.; and Selman, B. 1995. The comparative linguistics of knowledge representation. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 862–869. Morgan Kaufmann.

Guy, S., and Schwitter, R. 2016. The PENG^{ASP} system: Architecture, language and authoring tool. *Language Resources and Evaluation* 1–26.

Heyninck, J., and Straßer, C. 2016. Relations between assumption-based approaches in nonmonotonic logic and formal argumentation. In *Proceedings of NMR 2016*.

Kratzer, A. 2012. Modals and conditionals.

Kuhn, T. 2007. AceRules: Executing rules in controlled natural language. In Marchiori, M.; Pan, J. Z.; and de Sainte Marie, C., eds., *Web Reasoning and Rule Systems, First International Conference, RR 2007, Innsbruck , Austria, June 7-8, 2007, Proceedings*, volume 4524 of *Lecture Notes in Computer Science*, 299–308. Springer.

Kuhn, T. 2014. A survey and classification of controlled natural languages. *Computational Linguistics* 40(1):121–170.

Lierler, Y., and Lifschitz, V. 2013. Logic programs vs. first-order formulas in textual inference. *Computer Science Faculty Proceedings & Presentations*. Paper 16.

Lippi, M., and Torroni, P. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology* 16(2):10:1–10:25.

Pollock, J. L. 2007. Reasoning and probability. *Law, Probability and Risk* 6:43–58.

Poole, D. 1988. A logical framework for default reasoning. *Artificial Intelligence* 36(1):27–47.

Prakken, H. 2010. An abstract framework for argumentation with structured arguments. *Argument and Computation* 1(2):93–124.

Rescher, N., and Manor, R. 1970. On inferences from inconsistent premises. *Theory and Decision* 1(2):179–217.

Schulz, C., and Toni, F. 2016. Justifying answer sets using argumentation. *TPLP* 16(1):59–110.

Schulz, S. 2002. A comparison of different techniques for grounding near-propositional cnf formulae. In *Proceedings of the 15th International FLAIRS Conference*, 72–76.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529:484–489.

Strass, H. 2015. Instantiating rule-based defeasible theories in abstract dialectical frameworks and beyond. *Journal of Logic and Computation*. `http://dx.doi.org/10.1093/logcom/exv004`.

Toni, F. 2014. A tutorial on assumption-based argumentation. *Argument & Computation* 5(1):89–117.

Wyner, A.; Bench-Capon, T.; Dunne, P.; and Cerutti, F. 2015. Senses of 'argument' in instantiated argumentation frameworks. *Argument & Computation* 6(1):50–72.