# A Logic-based Framework for Decoding Enthymemes in Argument Maps involving Implicitness in Premises and Claims

**Victor David**[1] and **Anthony Hunter**[2]
[1]Université Côte d'Azur, Inria, CNRS, I3S, France
[2]University College London, UK
victor.david@inria.fr, anthony.hunter@ucl.ac.uk

## Abstract

Argument mining is a natural language processing technology aimed at identifying the explicit premises and claims of arguments in text, and the support and attack relationships between them. To better understand, and automatically analyse, the argument maps that are output from argument mining, it would be desirable to instantiate the arguments in the argument map with logical arguments. However, most real-world arguments are enthymemes (i.e. some of the premises and/or claims are implicit), which need to be decoded (i.e. the implicit aspects need to be identified). A key challenge is to decode enthymemes so as to respect the support and attack relationships in the argument map. To address this, we present a novel framework, based on default logic, for representing arguments including enthymemes. We show how decoding an enthymeme means identifying the default rules that are implicit in the premises and claims. We then show how choosing a decoding of the enthymemes in an argument map can be formalized as an optimization problem, and that a solution can be obtained using MaxSAT solvers.

## 1 Introduction

There are a number of frameworks for modelling argumentation in logic [Besnard *et al.*, 2014]. They incorporate a formal representation of individual arguments, where the premises imply the claims, and techniques for comparing conflicting arguments. However, real arguments presented by humans usually have insufficient explicit premises to logically infer the claims and/or insufficient explicit claims to logically support or attack the arguments that they are meant to. This is because the proponent of an argument assumes that the proponent and the intended recipient have some shared knowledge (common or commonsense), and this shared knowledge is often not presented explicitly in the argument. An argument with some implicit premises and/or claims is an **enthymeme** [Walton, 2001; Walton, 2019].

In this paper, we focus on enthymemes arising in argument maps. We assume an argument map is obtained as output from argument mining of a text, e.g. a discussion document, (see [Lawrence and Reed, 2019] for a review of argument mining). Each node in an argument map denotes an argument (often with some/all of the premises and/or claims being implicit) as illustrated in Figure 1.

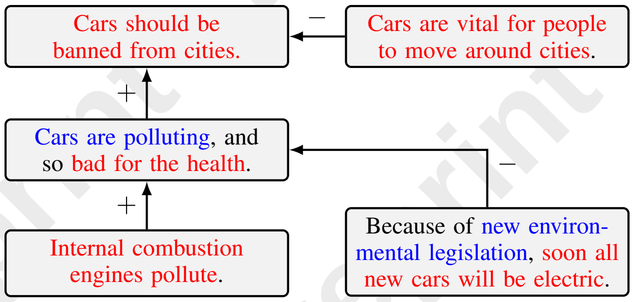

Cars should be banned from cities.
Cars are vital for people to move around cities.
Cars are polluting, and so bad for the health.
Because of new environmental legislation, soon all new cars will be electric.
Internal combustion engines pollute.
−
+
+
−

Figure 1: Example of an argument map (blue denotes premises, and red denotes claim) that might be obtained from the following text: *Should we ban cars from our cities? There is no doubt that the internal combustion engine is polluting. Hence, cars are polluting and so bad for the health. On the other hand, cars are vital for people to move around our cities, and because of new environmental legislation, soon all new cars will be electric.*

**Definition 1.** *Let $\mathcal{T}$ be a set of text strings (where each string is a phrase or sentence). An* **argument map** *is a tuple $(N, P, C, L)$ where $N$ is a set of nodes; $P : N \to \mathcal{T} \cup \{\text{Null}\}$ (resp. $C : N \to \mathcal{T} \cup \{\text{Null}\}$) is a text labelling function for premises (resp. claims); And $L : N \times N \to \{+, -, *\}$ is an arc labelling function, where $+$ (resp. $-$) represents a support (resp. attack) relationship, and $*$ represents no relationship holds.*

Human agents constantly need to understand enthymemes, whether in everyday or professional life, and so we need to replicate this process in computational models of argument. There are proposals for modelling decoding of enthymemes as abduction [Hunter, 2007; Black and Hunter, 2012; Hosseini *et al.*, 2014], and for how this can be undertaken within a dialogue [Black and Hunter, 2008; de Saint-Cyr, 2011b; de Saint-Cyr, 2011a; Xydis *et al.*, 2020; Panisson *et al.*, 2022; Leiva *et al.*, 2023; Xydis *et al.*, 2024]. However, most of these proposals only consider implicitness in premises, and none consider optimization of decoding.

To address these shortcomings, we propose a novel framework that has the following features: (1) A **representation**

**of explicit premises** (respectively **claims**) for each node of the argument map as a set of classical logic formulae for the premises (respectively claims) (2) A **representation of implicit premises** (respectively **claims**) for each node of the argument map as a set of default rules for the premises (respectively claims); (3) And a **mechanism for decoding enthymemes** (i.e. identifying the implicit premises and claims) based on maximizing the number of arcs between the logical arguments that agree with the corresponding label in the argument map. In other words, if the argument map has a support (respectively attack) relationship from node $n$ to node $m$, then the logical argument assigned to $n$ supports (respectively attacks) the logical argument assigned to $m$. Ideally, there is a (unique) decoding for each enthymeme that satisfies the labelling of arcs in the argument map. If not, we want to make the best compromise for the decoding of the enthymemes.

## 2 Background

We use $\mathcal{L}$ to denote the language of classical logic, $\alpha, \beta, \gamma, \delta, \ldots \in \mathcal{L}$ for arbitrary formulae of classical logic and $\Delta, \Gamma, \ldots \subseteq \mathcal{L}$ for arbitrary sets of classical formulae. Let $\top$ (respectively $\bot$) denote tautology (respectively contradiction), let $\vdash$ denote the classical consequence relation, and let $\mathsf{Cn}$ be the consequence closure function (i.e. $\mathsf{Cn}(\Delta) = \{\alpha \mid \Delta \vdash \alpha\}$). For $\alpha, \beta \in \mathcal{L}$, $\alpha \equiv \beta$ denotes that $\alpha$ and $\beta$ are equivalent (i.e. $\{\alpha\} \vdash \beta$ and $\{\beta\} \vdash \alpha$). For $\Delta, \Gamma \subseteq \mathcal{L}$, $\Delta \equiv \Gamma$ denotes that $\Delta$ and $\Gamma$ are equivalent (i.e. $\mathsf{Cn}(\Delta) = \mathsf{Cn}(\Gamma)$).

Our approach is based on default logic [Reiter, 1980]. It is one of the best known and most widely studied formalisations of default reasoning. It offers a very expressive and lucid language, and it captures various kinds of commonsense knowledge [Brewka, 1991; Davis, 2017] that are potentially important in representing implicit knowledge in enthymemes.

In default logic, knowledge is represented as a set of propositional or first-order formulae and a set of default rules for representing default information. A **default rule** is of the following form (which generalizes natural deduction rules), where $\alpha$, $\beta$ and $\gamma$ are classical formulae.

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

For this, $\alpha$ is the **pre-condition**, $\beta$ is the **justification**, and $\gamma$ is the **consequent**, of the default rule. For convenience, we may represent a default rule inline as $\alpha : \beta/\gamma$. Let $\mathcal{D}$ be the set of default rules.

A **default theory** is a pair $(W, D)$ where $W$ is a set of classical formulae and $D$ is a set of default rules. Default logic extends classical logic. Hence, all classical inferences from the classical information in a default theory are derivable (if there is an extension as defined below). The default theory then augments these classical inferences by default inferences derivable using the default rules: If $\alpha$ is inferred, and $\neg\beta$ cannot be inferred, then infer $\gamma$. The following is a definition for when a default theory has an extension (a set of classical logic formulae that follows from the default theory).

**Definition 2.** *Let $(W, D)$ be a default theory. The operator* $\mathsf{Pe}$ *identifies the conclusions for a given a set of classical formulae $E$. For this, $\mathsf{Pe}(E)$ is the smallest set of classical formulae s.t. the following three conditions are satisfied.*

1. $W \subseteq \mathsf{Pe}(E)$
2. $\mathsf{Pe}(E) = \mathsf{Cn}(\mathsf{Pe}(E))$
3. *For each default in $\alpha : \beta/\gamma \in D$, the following holds: if $\alpha \in \mathsf{Pe}(E)$, and $\neg\beta \notin E$, then $\gamma \in \mathsf{Pe}(E)$*

*We refer to $E$ as the* **satisfaction set**, *and $\mathsf{Pe}(E)$ as the* **potential extension**. *Furthermore, $E$ is an* **extension** *of $(W, D)$ iff $E = \mathsf{Pe}(E)$.*

For our definition for default arguments (Definition 4), we need the following subsidiary definition.

**Definition 3.** *A default theory $(W, D)$ is* **singular** *iff there is a unique extension of $(W, D)$. When a default theory $(W, D)$ is singular, let $\mathsf{Ex}(W, D)$ denote the extension.*

**Example 1.** *Let $W = \{\mathtt{a}\}$ and $D = \{(\mathtt{a} : \mathtt{b}/\mathtt{b}), (\mathtt{b} \lor \mathtt{c} : \mathtt{d} \land \mathtt{f}/\mathtt{e})\}$. The default theory $(W, D)$ is singular and its unique extension is $\mathsf{Ex}(W, D) = \mathsf{Cn}(\{\mathtt{a}, \mathtt{b}, \mathtt{e}\})$. In contrast, for $D' = (\{(\mathtt{a} : \mathtt{b}/\mathtt{b}), (\mathtt{a} : \neg\mathtt{b}/\neg\mathtt{b})\}$, $(D', W)$ is not singular, as there are two extensions $\mathsf{Cn}(\{\mathtt{a}, \mathtt{b}\})$ and $\mathsf{Cn}(\{\mathtt{a}, \neg\mathtt{b}\})$.*

Importantly for our purposes, any default theory with an extension $E$ can be turned into a singular default theory by removing defaults without changing the extension $E$.

**Proposition 1.** *For a default theory $(W, D)$, if $E$ is an extension of $(W, D)$, and $(W, D)$ is not singular, then there is a default theory $(W, D')$ s.t. $D' \subseteq D$ and $E$ is an extension of $(W, D')$, and $(W, D')$ is singular* [1].

**Example 2.** *Let $D = \{\mathtt{a} : \mathtt{b}/\mathtt{b}, \mathtt{a} : \neg\mathtt{b}/\neg\mathtt{b}\}$ and $W = \{\mathtt{a}\}$. So there are two extensions from $(W, D)$ which are $E_1 = \{\mathtt{a}, \mathtt{b}\}$ and $E_2 = \{\mathtt{a}, \neg\mathtt{b}\}$. The subtheory $(W, D_1)$ where $D_1 = \{\mathtt{a} : \mathtt{b}/\mathtt{b}\}$ is singular with the extension being $E_1$, and the subtheory $(W, D_2)$ where $D_2 = \{\mathtt{a} : \neg\mathtt{b}/\neg\mathtt{b}\}$ is singular with the extension being $E_2$.*

We use singular default theories in the definition of a default argument (Definition 4) to ensure that the implicit premises (respectively implicit claim) give a single perspective on the explicit premises (respectively explicit claim).

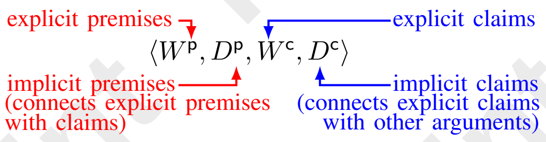

explicit premises
explicit claims
$\langle W^{\mathsf{p}}, D^{\mathsf{p}}, W^{\mathsf{c}}, D^{\mathsf{c}} \rangle$
implicit premises
(connects explicit premises with claims)
implicit claims
(connects explicit claims with other arguments)

Figure 2: Structure of a default argument where $(W^{\mathsf{p}}, D^{\mathsf{p}})$ and $(W^{\mathsf{c}}, D^{\mathsf{c}})$ are singular default theories.

## 3 Default Arguments

The following definition of default argument is very general. It allows us to represent enthymemes that may have insufficient premises to entail the explicit claim, and/or insufficient claims to attack or support arguments that it is meant to. We will consider constraints on the definition in order to give us appropriate notions of logical argument. We summarize the structure of a default argument in Figure 2.

[1] See appendix for proofs and code [David and Hunter, 2025]

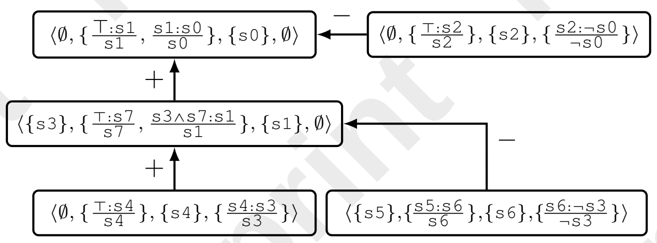

$\langle\emptyset, \{\frac{\top:\mathtt{s1}}{\mathtt{s1}}, \frac{\mathtt{s1}:\mathtt{s0}}{\mathtt{s0}}\}, \{\mathtt{s0}\}, \emptyset\rangle$
$-$
$\langle\emptyset, \{\frac{\top:\mathtt{s2}}{\mathtt{s2}}\}, \{\mathtt{s2}\}, \{\frac{\mathtt{s2}:\neg\mathtt{s0}}{\neg\mathtt{s0}}\}\rangle$
$+$
$\langle\{\mathtt{s3}\}, \{\frac{\top:\mathtt{s7}}{\mathtt{s7}}, \frac{\mathtt{s3}\wedge\mathtt{s7}:\mathtt{s1}}{\mathtt{s1}}\}, \{\mathtt{s1}\}, \emptyset\rangle$
$-$
$+$
$\langle\emptyset, \{\frac{\top:\mathtt{s4}}{\mathtt{s4}}\}, \{\mathtt{s4}\}, \{\frac{\mathtt{s4}:\mathtt{s3}}{\mathtt{s3}}\}\rangle$
$\langle\{\mathtt{s5}\}, \{\frac{\mathtt{s5}:\mathtt{s6}}{\mathtt{s6}}\}, \{\mathtt{s6}\}, \{\frac{\mathtt{s6}:\neg\mathtt{s3}}{\neg\mathtt{s3}}\}\rangle$

Figure 3: An instantiated argument map for the argument map given in Figure 1 where the atoms are $\mathtt{s0}$ = *cars should be banned from cities*, $\mathtt{s1}$ = *cars are bad for health*, $\mathtt{s2}$ = *cars are vital for people to move around cities*, $\mathtt{s3}$ = *cars are polluting*, $\mathtt{s4}$ = *internal combustion engines pollute*, $\mathtt{s5}$ = *new environmental legislation*, $\mathtt{s6}$ = *soon all cars will be electric*, and $\mathtt{s7}$ = *pollution is bad for health*.

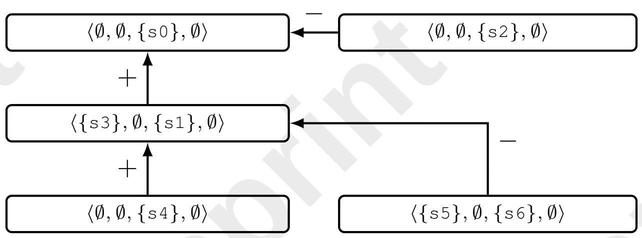

$\langle\emptyset, \emptyset, \{\mathtt{s0}\}, \emptyset\rangle$
$-$
$\langle\emptyset, \emptyset, \{\mathtt{s2}\}, \emptyset\rangle$
$+$
$\langle\{\mathtt{s3}\}, \emptyset, \{\mathtt{s1}\}, \emptyset\rangle$
$-$
$+$
$\langle\emptyset, \emptyset, \{\mathtt{s4}\}, \emptyset\rangle$
$\langle\{\mathtt{s5}\}, \emptyset, \{\mathtt{s6}\}, \emptyset\rangle$

Figure 4: An instantiated argument map for the argument map given in Figure 1 where each instantiated argument is a full enthymeme (using the atoms specified in Figure 3).

**Definition 4.** *A* **default argument** *is a tuple* $\langle W^{\mathsf{p}}, D^{\mathsf{p}}, W^{\mathsf{c}}, D^{\mathsf{c}}\rangle$ *where* $W^{\mathsf{p}}, W^{\mathsf{c}} \subseteq \mathcal{L}$, *and* $D^{\mathsf{p}}, D^{\mathsf{c}} \subseteq \mathcal{D}$ *such that* $(W^{\mathsf{p}}, D^{\mathsf{p}})$ *is singular and* $(W^{\mathsf{c}}, D^{\mathsf{c}})$ *is singular.*

For a default argument $A = \langle W^{\mathsf{p}}, D^{\mathsf{p}}, W^{\mathsf{c}}, D^{\mathsf{c}}\rangle$, we refer to $W^{\mathsf{p}}$ as the **explicit premises**, $D^{\mathsf{p}}$ as the **implicit premises**, $W^{\mathsf{c}}$ as the **explicit claims**, and $D^{\mathsf{c}}$ as the **implicit claims**. To extract these components, we use the following functions: $\mathsf{Ep}(A) = W^{\mathsf{p}}$; $\mathsf{Ip}(A) = D^{\mathsf{p}}$; $\mathsf{Ec}(A) = W^{\mathsf{c}}$; and $\mathsf{Ic}(A) = D^{\mathsf{c}}$.

**Example 3.** *The following are examples of default arguments:*

$\mathtt{A} = \langle\{\mathtt{a}\vee\mathtt{b}\}, \{(\mathtt{a}\vee\mathtt{b}\vee\mathtt{c} : \mathtt{d}/\mathtt{d})\}, \{\mathtt{d}\}, \{(\mathtt{d} : \neg\mathtt{e}/\neg\mathtt{e})\}\rangle$.
$\mathtt{B} = \langle\emptyset, \emptyset, \{\mathtt{b}\vee\neg\mathtt{b}\}, \emptyset\rangle$.
$\mathtt{C} = \langle\emptyset, \{(\top : \mathtt{e}/\mathtt{e})\}, \{\mathtt{e}\}, \{(\mathtt{e} : \mathtt{f}/\mathtt{f}), (\mathtt{f} : \mathtt{g}\wedge\mathtt{h}/\mathtt{h})\}\rangle$.

Given a default argument $A$, the support of the argument is the default extension obtained from the implicit and explicit premises, and the consequence of the argument is the default extension obtained from the implicit and explicit claims:

- The **support** of $A$ is $\mathsf{S}(A) = \mathsf{Ex}(\mathsf{Ep}(A), \mathsf{Ip}(A))$.
- The **consequence** of $A$ is $\mathsf{C}(A) = \mathsf{Ex}(\mathsf{Ec}(A), \mathsf{Ic}(A))$.

So the support is the set of inferences from the implicit and explicit premises and the consequence is the set inferences from the implicit and explicit claims.

**Example 4.** *Continuing Example 3, for the default argument* $\mathtt{A}$ *to* $\mathtt{C}$, *we have the following support and consequence.*

| | |
|---|---|
| $\mathsf{S}(\mathtt{A}) = \mathsf{Cn}(\{\mathtt{a}\vee\mathtt{b}, \mathtt{d}\})$ | $\mathsf{C}(\mathtt{A}) = \mathsf{Cn}(\{\mathtt{d}, \neg\mathtt{e}\})$ |
| $\mathsf{S}(\mathtt{B}) = \mathsf{Cn}(\emptyset)$ | $\mathsf{C}(\mathtt{B}) = \mathsf{Cn}(\emptyset)$ |
| $\mathsf{S}(\mathtt{C}) = \mathsf{Cn}(\{\mathtt{e}\})$ | $\mathsf{C}(\mathtt{C}) = \mathsf{Cn}(\{\mathtt{e}, \mathtt{f}, \mathtt{h}\})$ |

We explain the definition below as follows: A default argument is valid iff the explicit claims are in the support of the argument (i.e. the extension of the premises); A default argument is implicit premise minimal iff there is no subset of the implicit premises such that the argument is valid; And a default argument is support (respectively consequence) consistent iff the support (respectively consequence) is consistent.

**Definition 5.** *For default argument $A$: $A$ is* **valid** *iff* $\mathsf{Ec}(A) \subseteq \mathsf{S}(A)$; *$A$ is* **implicit premise minimal** *iff $A$ is valid and there is no* $D' \subset \mathsf{Ip}(A)$ *s.t.* $\mathsf{Ec}(A) \subseteq \mathsf{Ex}(\mathsf{Ep}(A), D')$; *and $A$ is* **support consistent** *(respectively* **consequence consistent***) iff* $\bot \notin \mathsf{S}(A)$ *(respectively* $\bot \notin \mathsf{C}(A)$*).*

**Example 5.** *For Example 3,* $\mathtt{A}$. $\mathtt{B}$, *and* $\mathtt{C}$, *are valid, implicit premise minimal, and support and consequence consistent.*

A property of default logic is that an extension of a default theory $(W, D)$ is inconsistent iff $W$ is inconsistent. So for a default argument, the support (respectively consequence) is inconsistent iff the explicit premises (respectively explicit consequence) are inconsistent. Hence, the implicit premises (respectively implicit claims) cannot cause the support (respectively consequence) to be inconsistent.

**Example 6.** *For* $A = \langle\{\mathtt{a}\wedge\neg\mathtt{a}\}, \emptyset, \{\mathtt{d}\}, \{\mathtt{d} : \mathtt{e}/\mathtt{e}\}\rangle$ *is such that* $\bot \in \mathsf{S}(A)$ *and* $\bot \notin \mathsf{C}(A)$, *whereas* $B = \langle\{\mathtt{a}\}, \{\mathtt{a} : \mathtt{b}/\mathtt{b}\}, \{\mathtt{d}\wedge\neg\mathtt{d}\}, \emptyset\rangle$ *is such that* $\bot \notin \mathsf{S}(B)$ *and* $\bot \in \mathsf{C}(B)$.

Given an argument map, and a set of default arguments, we define a default argument assignment as follows, and illustrate this in Figures 3 and 4.

**Definition 6.** *Let* $M = (N, P, C, L)$ *be an argument map and let $\mathcal{A}$ be a set of default arguments. A* **default argument assignment** *for $M$ is a function* $I : N \to \mathcal{A}$.

We refer to an argument map with a default argument assigned to each node as an **instantiated argument map** (as illustrated in Figures 3 and 4). So each default argument assignment can be presented as an instantiated argument map.

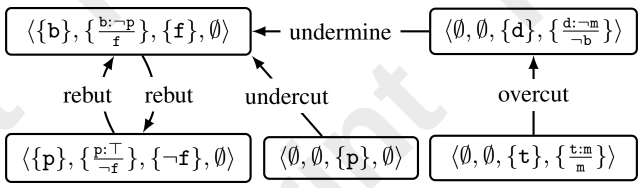
We now consider some of the options for defining one argument supporting another.

**Definition 7.** *For default arguments $A$ and $B$, the following are some definitions for $A$ supporting $B$: $A$* **explicit premise supports** *$B$ iff* $\mathsf{C}(A)\cap\mathsf{Ep}(B) \neq \emptyset$; *$A$* **explicit claim supports** *$B$ iff* $\mathsf{C}(A) \cap \mathsf{Ec}(B) \neq \emptyset$; *$A$* **premise justification supports** *$B$ iff there exists* $\beta \in \mathsf{C}(A)$ *s.t. there exists* $\alpha : \beta/\gamma \in \mathsf{Ip}(B)$; *and $A$* **claim justification supports** *$B$ iff there exists* $\beta \in \mathsf{C}(A)$ *s.t. there exists* $\alpha : \beta/\gamma \in \mathsf{Ic}(B)$.

**Example 7.** *Consider the following supporting default arguments (left) and supported default arguments (right):* $\mathtt{A1}$ *explicit premise supports* $\mathtt{B1}$*;* $\mathtt{A2}$ *explicit claim supports* $\mathtt{B2}$*;* $\mathtt{A3}$ *premise justification supports* $\mathtt{B3}$*; and* $\mathtt{A4}$ *claim justification supports* $\mathtt{B4}$.

| | |
|---|---|
| $\mathtt{A1}=\langle\emptyset,\emptyset,\{\mathtt{a}\},\{\mathtt{a}:\mathtt{b}/\mathtt{b}\}\rangle$ | $\mathtt{B1}=\langle\{\mathtt{b}\},\{\mathtt{b}:\mathtt{c}/\mathtt{c}\},\{\mathtt{c}\},\emptyset\rangle$ |
| $\mathtt{A2}=\langle\{\mathtt{d}\},\{\mathtt{d}:\mathtt{a}/\mathtt{a}\},\{\mathtt{a}\},\emptyset\rangle$ | $\mathtt{B2}=\langle\{\mathtt{b}\},\{\mathtt{b}:\mathtt{c}/\mathtt{a}\},\{\mathtt{a}\},\emptyset\rangle$ |
| $\mathtt{A3}=\langle\emptyset,\emptyset,\{\mathtt{a}\},\emptyset\rangle$ | $\mathtt{B3}=\langle\{\mathtt{b}\},\{\mathtt{b}:\mathtt{a}/\mathtt{c}\},\{\mathtt{c}\},\emptyset\rangle$ |
| $\mathtt{A4}=\langle\emptyset,\emptyset,\{\mathtt{a}\},\{\mathtt{a}:\mathtt{b}/\mathtt{b}\}\rangle$ | $\mathtt{B4}=\langle\emptyset,\emptyset,\{\mathtt{c}\},\{\mathtt{c}:\mathtt{b}/\mathtt{b}\}\rangle$ |

Next, we consider some options for defining one argument attacking another and illustrate them in Figure 5.



$\langle\{\mathtt{b}\},\{\frac{\mathtt{b}:\neg\mathtt{p}}{\mathtt{f}}\},\{\mathtt{f}\},\emptyset\rangle$
undermine
$\langle\emptyset,\emptyset,\{\mathtt{d}\},\{\frac{\mathtt{d}:\neg\mathtt{m}}{\neg\mathtt{b}}\}\rangle$
rebut
rebut
undercut
overcut
$\langle\{\mathtt{p}\},\{\frac{\mathtt{p}:\top}{\neg\mathtt{f}}\},\{\neg\mathtt{f}\},\emptyset\rangle$
$\langle\emptyset,\emptyset,\{\mathtt{p}\},\emptyset\rangle$
$\langle\emptyset,\emptyset,\{\mathtt{t}\},\{\frac{\mathtt{t}:\mathtt{m}}{\mathtt{m}}\}\rangle$

Figure 5: An instantiated argument map where $\mathtt{b}$ = *it is a bird*, $\mathtt{p}$ = *it is a penguin*, $\mathtt{d}$ = *it is a decoy model that looks like a bird* (i.e. a realistic model of a game bird used by hunters to lure prey to the hunter's position), $\mathtt{t}$ = *it is twitching*, $\mathtt{m}$ = *it is moving like a bird*, and $\mathtt{f}$ = *it is capable of flying*.

**Definition 8.** *For default arguments $A$ and $B$, the following are some definitions for $A$ attacking $B$: $A$* **undermines** *$B$ iff there exists $\neg\beta \in \mathsf{C}(A)$ s.t. $\beta \in \mathsf{Ep}(B)$; $A$* **rebuts** *$B$ iff there exists $\neg\beta \in \mathsf{C}(A)$ s.t. $\beta \in \mathsf{Ec}(B)$; $A$* **undercuts** *$B$ iff there exists $\neg\beta \in \mathsf{C}(A)$ s.t. there exists $\alpha : \beta/\gamma \in \mathsf{Ip}(B)$; and $A$* **overcuts** *$B$ iff there exists $\neg\beta \in \mathsf{C}(A)$ s.t. there exists $\alpha : \beta/\gamma \in \mathsf{Ic}(B)$.*

A default argument provides a richer representation of an argument than available with other approaches to structured argumentation. As illustrated in Figure 2, this includes the following features which together go beyond other formalisms: (1) Delineation of implicit information connecting premises and claims (i.e. the set $D^{\mathsf{p}}$ is a set of defaults that represents the implicit information in the premises); (2) Logical mechanism for disabling connection between premises and claims (i.e. the justification of each default rule can be negated by the claims of another argument, thereby attacking the connection between the premises and claims); (3) Delineation of implicit information connecting one argument with another (i.e. the set $D^{\mathsf{c}}$ is a set of defaults that represents the implicit information in the claims); (4) Logical mechanism for disabling connection between one argument and another (i.e. the justification of each default rule can be negated by the claims of another argument, thereby attacking the connection between that argument and other arguments).

Definitions 7 and 8 are only some of the possible definitions for support and attack. Whatever definition we use for support and attack, if $\mathcal{A}$ is the set of logical arguments, then $R^+ \subseteq (\mathcal{A} \times \mathcal{A})$ (respectively $R^- \subseteq (\mathcal{A} \times \mathcal{A})$) denotes the set of pairs of logical arguments $(A, B)$ where $A$ supports (respectively attacks) $B$. Also, let $R^*$ be the set of pairs of logical argument where $A$ does not support nor attack $B$ (i.e., $R^* = (\mathcal{A} \times \mathcal{A}) \setminus (R^+ \cup R^-)$).

The definition for a default argument is very general. As discussed in the next section, it allows for the representation and decoding of enthymemes. We leave to further work the presentation of further interesting classes of default argument, and of further types of attack and support relationship, that can be captured in this framework.

## 4 Decoding Enthymemes

An enthymeme is an argument with missing premises and/or claims. We start by considering types of enthymeme involving missing premises.

**Definition 9.** *A default argument $A$ is a* **premise enthymeme** *iff $A$ is not valid. Additionally, if $A$ is a premise enthymeme and $\mathsf{Ip}(A) = \emptyset$, then $A$ is an* **acute premise enthymeme***, and if $A$ is a premise enthymeme and $\mathsf{Ip}(A) = \emptyset$ and $\mathsf{Ic}(A) = \emptyset$, then $A$ is a* **full enthymeme***.*

To illustrate, each default argument in the instantiated argument map in Figure 4 is a full enthymeme.

When we have an argument map $M = (N, P, C, L)$, we assume that for each node $n \in N$, we can use natural language processing on the text $P(n)$ and $C(n)$ to obtain a full enthymeme for that node. So each explicit premise and explicit claim is obtained by translating the text into formulae of classical logic. This gives us a default argument assignment $I_e$ as an **initial enthymeme assignment** where every default argument that is assigned is a full enthymeme (as illustrated in the instantiated argument map in Figure 4).

Next, we consider types of enthymeme involving missing claims. This is more complicated as we need to consider the relationship of an argument with other arguments.

**Definition 10.** *Let $R^+$ (resp. $R^-$) be a support (resp. attack) relation. For $A, B \in \mathcal{A}$, $A$ is a* **support enthymeme** *(resp.* **attack enthymeme***) to $B$ iff $(A, B) \notin R^+$ (resp. $(A, B) \notin R^-$). Additionally, if $A$ is a support enthymeme (resp. attack enthymeme) to $B$, and $\mathsf{Ic}(A) = \emptyset$, then $A$ is an* **acute support enthymeme** *(resp.* **acute attack enthymeme***) to $B$. If $A$ is a support enthymeme, or attack enthymeme, to $B$, then $A$ is a* **claim enthymeme** *to $B$.*

To illustrate, each default argument in Figure 4 is an acute support enthymeme, and an acute attack enthymeme, to every other default argument in this figure. In this example, no default argument supports or attacks any other.

Next, we consider decoding a premise enthymeme as an abduction problem (drawing on [Eiter *et al.*, 1997]). For this, we assume a **default knowledgebase**, denoted $K$, which is a set of default rules. For premise enthymeme $A$, we want to find a subset of $K$ to add to $\mathsf{Ip}(A)$ for it to be valid as follows.

**Proposition 2.** *For default knowledgebase $K$, the problem of identifying a default argument $C$ for a premise enthymeme $A$ s.t. (1) $\mathsf{Ip}(C) \setminus \mathsf{Ip}(A) \subseteq K$, (2) $\mathsf{Ep}(A) = \mathsf{Ep}(C)$, (3) $\mathsf{Ec}(A) = \mathsf{Ec}(C)$, and (4) $C$ is valid, is in $\Sigma_2^{\mathsf{P}}$.*

The above result does not consider implicitness in the claim: If we have a default argument $A$ that is a support enthymeme, or attack enthymeme, for a default argument $B$, then we need to take $B$ into account in the decoding.

**Proposition 3.** *Let $R^+$ (resp. $R^-$) be a support (resp. attack) relation according to Definition 7 (resp. Definition 8). Let $A$ be a support enthymeme (resp. attack enthymeme) to $B$. For default knowledgebase $K$, the problem of identifying a default argument $C$ s.t. (1) $\mathsf{Ic}(C) \setminus \mathsf{Ic}(A) \subseteq K$, (2) $\mathsf{Ep}(A) = \mathsf{Ep}(C)$, (3) $\mathsf{Ec}(A) = \mathsf{Ec}(C)$, and (4) $(C, B) \in R^+$ (respectively $(C, B) \in R^-$), is in $\Sigma_2^{\mathsf{P}}$.*

We want to use the argument map to determine which arguments are claim enthymemes that we need to decode accordingly: If according to the argument map, $L(n, m) = +$ (respectively $L(n, m) = -$) holds for a pair of nodes $n$ and $m$, and the default argument $A$ assigned to $n$, and the default argument $B$ assigned to $m$, are such that $A$ is a support enthymeme (respectively attack enthymeme) to $B$, then

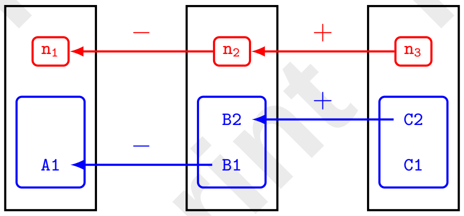


Figure 6: Schematic representation of the choices for the default argument assignment. The structure of an argument map is given by the nodes $\mathtt{n}_1$, $\mathtt{n}_2$, and $\mathtt{n}_3$. Suppose for $\mathtt{n}_1$ (respectively $\mathtt{n}_2$ and $\mathtt{n}_3$), we can assign A1 (respectively $\mathtt{B}_1$ or $\mathtt{B}_2$ and $\mathtt{C}_1$ or $\mathtt{C}_2$). If we choose A1 and B1, we get the attack from $\mathtt{n}_2$ to $\mathtt{n}_1$, but then there is no assignment for $\mathtt{n}_3$ to give an attack from $\mathtt{n}_3$ to $\mathtt{n}_2$. Alternatively, if we choose B2 and C2, we get the support from $\mathtt{n}_3$ to $\mathtt{n}_2$, but then there is no assignment for $\mathtt{n}_1$ to give an attack from $\mathtt{n}_2$ to $\mathtt{n}_1$.

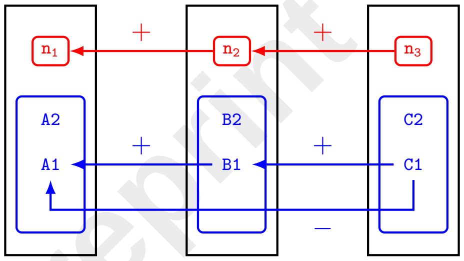


Figure 7: Schematic representation of the choices for default argument assignment for the argument map with nodes $\mathtt{n}_1$, $\mathtt{n}_2$, and $\mathtt{n}_3$. Suppose we choose A1 and B1, then we satisfy the support from $\mathtt{n}_2$ to $\mathtt{n}_1$, but then for $\mathtt{n}_3$, we either choose C1 to satisfy support from $\mathtt{n}_3$ to $\mathtt{n}_1$ but with an undesired attack from $\mathtt{n}_3$ to $\mathtt{n}_1$, or we choose C2 and thereby fail to satisfy the support from $\mathtt{n}_3$ to $\mathtt{n}_2$. Alternatively, we could choose A2, and so fails to satisfy the support from $\mathtt{n}_2$ to $\mathtt{n}_1$.

we want to decode $A$ by adding default rules to the implicit claims of $A$ so that it then supports (respectively attacks) $B$ accordingly.

A further issue is that the choice of implicit premises for each argument can affect what is possible for the choice of implicit claims for the other arguments, as illustrated next.

**Example 8.** *Consider the initial enthymeme assignment where the left (resp. right) node is* $\mathtt{n}_1$ *(resp.* $\mathtt{n}_2$*). From default knowledgebase* $K = \{(\mathtt{a1} \wedge \mathtt{a2} : \mathtt{a3}/\mathtt{a4}), (\mathtt{a1} \vee \mathtt{a2} : \mathtt{a4}/\mathtt{a4}), (\top : \mathtt{b1}/\mathtt{b1}), (\mathtt{b1} \vee \mathtt{b2} : \neg\mathtt{a3}/\neg\mathtt{a3})\}$*, suppose we assign* $\mathtt{A1} = \langle\{\mathtt{a1},\mathtt{a2}\}, \{(\mathtt{a1} \vee \mathtt{a2} : \mathtt{a4}/\mathtt{a4})\}, \{\mathtt{a4}\}, \emptyset\rangle$ *to* $\mathtt{n}_1$ *and* $\mathtt{A2} = \langle\emptyset, \{(\top : \mathtt{b1}/\mathtt{b1})\}, \{\mathtt{b1}\}, \{(\mathtt{b1} \vee \mathtt{b2} : \neg\mathtt{a3}/\neg\mathtt{a3})\}\rangle$ *to* $\mathtt{n}_2$*, then* A2 *does not undercut* A1*. But, we can assign* $\mathtt{A3} = \langle\{\mathtt{a1},\mathtt{a2}\}, \{(\mathtt{a1} \wedge \mathtt{a2} : \mathtt{a3}/\mathtt{a4})\}, \{\mathtt{a4}\}, \emptyset\rangle$ *to* $\mathtt{n}_1$*, and then* A2 *does undercut* A3*.*

$$\begin{tikzcd} \boxed{\mathtt{n}_1 \quad \langle\{\mathtt{a1},\mathtt{a2}\},\emptyset,\{\mathtt{a4}\},\emptyset\rangle} & \boxed{\mathtt{n}_2 \quad \langle\emptyset,\emptyset,\{\mathtt{b1}\},\emptyset\rangle} \arrow[l, "-"'] \end{tikzcd}$$

To address the issues raised above, we treat decoding of enthymemes as an optimization problem, in the next section.

## 5 Optimization of Decoding

Ideally, for an argument map $M = (N, P, C, L)$, and an initial enthymeme assignment $I_e$, we want to find a default argument assignment $I$ that respects the labelling of arcs (i.e. $L$). This means that for each arc $(n, m)$ in the argument map, and for each label $x \in \{+, -, *\}$, the following holds $L(n, m) = x$ iff $((I(n), I(m)) \in R^x)$. So the assignment $I$ would ensure that there is only attack (respectively support) between two default arguments when there is an attack (respectively support) between the corresponding nodes in the argument map, and there is no relation otherwise.

The ideal situation described above cannot be guaranteed in practice. We may be unable to choose a default argument assignment that respects all the arcs in the argument map (as illustrated in Figure 6, where we fail to obtain either an attack or support, and in Figure 7 where we get the required support relations, but also an undesired attack).

We say $B$ is a **decoding candidate** of full enthymeme $A$ iff $B$ is valid and $\mathsf{Ep}(A) = \mathsf{Ep}(A)$ and $\mathsf{Ec}(A) = \mathsf{Ec}(B)$. If there is a decoding candidate $B$ of full enthymeme $A$, let $\mathsf{Cands}(K, A)$ be the **set of decoding candidates** of $A$ from default knowledgebase $K$, otherwise let $\mathsf{Cands}(K, A) = \{A\}$. For an initial enthymeme assignment $I_e$, a **decoding assignment** $I_d$ is such that, for each $n \in N$, $I_d(n) \in \mathsf{Cands}(K, I_e(n))$. For an argument map $M$ and a default knowledge base $K$, let $\mathsf{Assigns}(M, K)$ be the **set of decodings assignments** for $M$ given $K$.

We score each decoding assignment $I_d \in \mathsf{Assigns}(M, K)$ in terms of the proportion of pairs of nodes in the argument map that are correctly identified as support, attack, or no relationship. First, we define the following set of pairs of nodes that are correctly assigned by $I_d$ according to the labelling function $L$: $\mathsf{Correct}(R^+, R^-, I_d, L) = \{(n_i, n_j) \mid$ for all $x \in \{+, -, *\}$, if $(I_d(n_i), I_d(n_j)) \in R^x$, then $L(n_i, n_j) = x\}$. So $0 \leq |\mathsf{Correct}(R^+, R^-, I_d, L)| \leq |N \times N|$. Then, the accuracy of each assignment $I_d$, is defined as follows: If $\mathsf{Correct}(R^+, R^-, I_d, L) = \emptyset$, then $\mathsf{Acc}(R^+, R^-, I_d, N, L) = 0$, otherwise, $\mathsf{Acc}(R^+, R^-, I_d, N, L) =$

$$\frac{|\mathsf{Correct}(R^+, R^-, I_d, L)|}{|N \times N|} \tag{1}$$

The decoding optimization problem is choosing the decoding assignment $I_d$ that maximizes the number of arcs correct w.r.t. to the labelling. For example, for Figure 6, an optimal decoding assignment is $I(\mathtt{n}_1) = \mathtt{A1}$, $I(\mathtt{n}_2) = \mathtt{B1}$, and $I(\mathtt{n}_3) = \mathtt{C1}$, which has one of two arcs correct w.r.t. the labelling in the argument map. So the **decoding optimization problem** is defined as follows where an optimal solution is a decoding assignment $I_d$ with maximum accuracy.

$$\arg \max_{I_d \in \mathsf{Assigns}(M,K)} \mathsf{Acc}(R^+, R^-, I_d, N, L) \tag{2}$$

Unfortunately, using the above formula to the optimal decoding assignments is impractical in general since it would involve generating all the decoding assignments. For an argument map $M$, default knowledgebase $K$, and initial enthymeme assignment $I_e$, the cardinality of $\mathsf{Assigns}(M, K)$ is $\prod_{n \in N} |\mathsf{Cands}(K, I_e(n))|$, and so if there are many decoding candidates per node, the number of decoding assignments would soon be unmanageable.

To address this problem, we show how we can reduce this optimization problem to a partial MaxSAT problem. Our approach is to translate the possible decoding assignments of the argument map into a set of hard constraints, and then let the choice of arcs be managed by soft constraints. The aim is to maximize the number of arcs that are consistent with the constraints of the possible decoding assignments. For this, we need a set of atoms from which to construct the constraints. Given a default knowledge base $K$ and enthymeme $A$, each element of $\mathsf{Cands}(K, A)$ is treated as an atom, which we refer to as a **candidate atom**. To illustrate, for Figure 6, the set of candidate atoms is $\{\mathtt{A1}, \mathtt{B1}, \mathtt{B2}, \mathtt{C1}, \mathtt{C2}\}$.

We also need atoms to represent information about the relationships between decoding candidates for each enthymeme $A$. For default arguments $B$ and $C$, we require atoms of the form $U^B_C$ which represents $B \in \mathsf{Cands}(K, I_e(n))$, and $C \in \mathsf{Cands}(K, I_e(m))$, and $x \in \{+, -, *\}$, and $(B, C) \in R^x$, then $L(n, m) \neq x$. In other words, $U^B_C$ denotes that for the candidate $B$ (respectively $C$) for a node $n$ (respectively $m$), the logical relationship from $B$ to $C$ disagrees with the label on the arc from $n$ to $m$. We refer to each $U^B_C$ atom as an **undesired atom**. To illustrate, for Figure 6, the set of undesired atoms is $\{\mathtt{U}^{\mathtt{B2}}_{\mathtt{A1}}, \mathtt{U}^{\mathtt{C1}}_{\mathtt{B1}}, \mathtt{U}^{\mathtt{C1}}_{\mathtt{B2}}$, and $\mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}}\}$.

For support relation $R^+$, attack relation $R^-$, default knowledgebase $K$, initial enthymeme assignment $I_e$, and labelling function $L$, let $\mathsf{Undes}(R^+, R^-, K, I_e, L)$ denote the set of undesired atoms (i.e. $\{U^B_C \mid$ for all $x \in \{+, -, *\}$, if there are $n, m \in N$ such that $B \in \mathsf{Cands}(K, I_e(n))$, and $C \in \mathsf{Cands}(K, I_e(m))$, and $(B, C) \in R^x$, then $L(n, m) \neq x\}$).

For simplifying the presentation in this paper, we assume the sets $R^+$, $R^-$, and $R^*$, are disjoint. In other words, we assume that there are no decoding candidates $B$ and $C$ such that $B$ is both an attacker and a supporter of $C$. However, in real-world argumentation, this is a possibility, and the methods in this paper can easily be generalized to deal with this.

Given a set of candidate atoms and a set of undesired atoms, we define the constraints as follows.

**Definition 11.** *For* $\mathsf{Cands}(K, I_e(n)) = \{B_1, \ldots, B_k\}$, *where* $n \in N$, *and for each* $U^B_C \in \mathsf{Undes}(R^+, R^-, K, I_e, L)$, *Hard1 to Hard3 are hard constraints, and Soft1 is a soft constraint:*

- **Hard1.** $B_1 \vee \ldots \vee B_k$
- **Hard2.** $\neg B_i \vee \neg B_j$ *for* $i, j \in \{1, \ldots, k\}$ *s.t.* $i \neq j$
- **Hard3.** $\neg C \vee \neg B \vee U^B_C$
- **Soft1.** $\neg U^B_C$

We explain these constraints as follows: (Hard1) For every node in the argument map, one of the candidate atoms has to be true (i.e. there is an assignment of a decoding candidate for each node); (Hard2) For every node, at most one of the candidates atoms for that node can be true (i.e. there cannot be multiple decoding candidates assigned to the same node); and (Hard3) For every undesired atom, either the undesired atom is true, or one of the candidate atoms associated with that undesired atom is false (i.e. either there is an arc in argument map that has a label that does not agree with the corresponding relationship between the two decoding candidates, or one of those decoding candidates is false); (Soft1) Each soft constraint is the negation of an undesired atom, and so satisfying a soft constraint means making the undesired atom false. Ideally, we want a model that satisfies all four types of constraint. If this is not possible, we seek a model of the hard constraints, and as many of the soft constraints as possible. Hence, we are seeking to maximize the number of undesired atoms that are false.

Given an argument map, and a set of decoding candidates for each node, we obtain the constraints as specified above. These are then given to a partial MaxSAT solver which then finds the models that satisfy the hard constraints, and maximizes the number of soft constraints satisfied. Each of these models specifies a decoding assignment. For a model, each candidate atom true in the model denotes the decoding candidate that is assigned to a node in the argument map by the decoding assignment.

**Example 9.** *Consider Figure 6 where the decoding candidates for* $\mathtt{n}_1$ *(respectively* $\mathtt{n}_2$ *and* $\mathtt{n}_3$*) are* $\{\mathtt{A1}\}$, *(respectively* $\{\mathtt{B1}, \mathtt{B2}\}$, *and* $\{\mathtt{C1}, \mathtt{C2}\}$*), where* $R^+ = \{(\mathtt{B2}, \mathtt{A2})\}$ *and* $R^- = \{(\mathtt{C1}, \mathtt{B1})\}$. *The hard constraints are as follows:*

$$\begin{array}{ccc}
\mathtt{A1} & \mathtt{B1} \vee \mathtt{B2} & \mathtt{C1} \vee \mathtt{C2} \\
\neg\mathtt{B1} \vee \neg\mathtt{B2} & \neg\mathtt{C1} \vee \neg\mathtt{C2} & \neg\mathtt{A1} \vee \neg\mathtt{B2} \vee \mathtt{U}^{\mathtt{B2}}_{\mathtt{A1}} \\
\neg\mathtt{B1} \vee \neg\mathtt{C1} \vee \mathtt{U}^{\mathtt{C1}}_{\mathtt{B1}} & \neg\mathtt{B2} \vee \neg\mathtt{C1} \vee \mathtt{U}^{\mathtt{C1}}_{\mathtt{B2}} & \neg\mathtt{B1} \vee \neg\mathtt{C2} \vee \mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}}
\end{array}$$

*The soft constraints are* $\{\neg\mathtt{U}^{\mathtt{B2}}_{\mathtt{A1}}, \neg\mathtt{U}^{\mathtt{C1}}_{\mathtt{B1}}, \neg\mathtt{U}^{\mathtt{C1}}_{\mathtt{B2}}, \neg\mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}}\}$. *The three optimal MaxSAT models are* $\{\mathtt{A1}, \mathtt{B2}, \mathtt{C2}, \mathtt{U}^{\mathtt{B2}}_{\mathtt{A1}}\}$, $\{\mathtt{A1}, \mathtt{B1}, \mathtt{C1}, \mathtt{U}^{\mathtt{C1}}_{\mathtt{B1}}\}$, *and* $\{\mathtt{A1}, \mathtt{B1}, \mathtt{C2}, \mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}}\}$, *giving three optimal decoding assignments* $I_1$, $I_2$, *and* $I_3$ *as follows:* $I_1(\mathtt{n}_1) = \mathtt{A1}$, $I_1(\mathtt{n}_2) = \mathtt{B2}$, $I_1(\mathtt{n}_3) = \mathtt{C2}$, $I_2(\mathtt{n}_1) = \mathtt{A1}$, $I_2(\mathtt{n}_2) = \mathtt{B1}$, $I_2(\mathtt{n}_3) = \mathtt{C1}$, $I_3(\mathtt{n}_1) = \mathtt{A1}$, $I_3(\mathtt{n}_2) = \mathtt{B1}$, *and* $I_3(\mathtt{n}_3) = \mathtt{C2}$.

**Example 10.** *Consider Figure 7 where the decoding candidates for* $\mathtt{n}_1$ *(respectively* $\mathtt{n}_2$ *and* $\mathtt{n}_3$*) are* $\{\mathtt{A1}, \mathtt{A2}\}$, *(respectively* $\{\mathtt{B1}, \mathtt{B2}\}$, *and* $\{\mathtt{C1}, \mathtt{C2}\}$*), and where* $R^+ = \{(\mathtt{B1}, \mathtt{A1}), (\mathtt{C1}, \mathtt{B1})\}$ *and* $R^- = \{(\mathtt{C1}, \mathtt{A1})\}$. *So we have the following formulae as hard constraints,*

$$\begin{array}{ccc}
\mathtt{A1} \vee \mathtt{A2} & \mathtt{B1} \vee \mathtt{B2} & \mathtt{C1} \vee \mathtt{C2} \\
\neg\mathtt{A1} \vee \neg\mathtt{A2} & \neg\mathtt{B1} \vee \neg\mathtt{B2} & \neg\mathtt{C1} \vee \neg\mathtt{C2} \\
\neg\mathtt{A2} \vee \neg\mathtt{B1} \vee \mathtt{U}^{\mathtt{B1}}_{\mathtt{A2}} & \neg\mathtt{A1} \vee \neg\mathtt{B2} \vee \mathtt{U}^{\mathtt{B2}}_{\mathtt{A1}} & \neg\mathtt{A2} \vee \neg\mathtt{B2} \vee \mathtt{U}^{\mathtt{B2}}_{\mathtt{A2}} \\
\neg\mathtt{B2} \vee \neg\mathtt{C1} \vee \mathtt{U}^{\mathtt{C1}}_{\mathtt{B2}} & \neg\mathtt{B1} \vee \neg\mathtt{C2} \vee \mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}} & \neg\mathtt{B2} \vee \neg\mathtt{C2} \vee \mathtt{U}^{\mathtt{C2}}_{\mathtt{B2}} \\
 & \neg\mathtt{A1} \vee \neg\mathtt{C1} \vee \mathtt{U}^{\mathtt{C1}}_{\mathtt{A1}} &
\end{array}$$

*and* $\{\neg\mathtt{U}^{\mathtt{B1}}_{\mathtt{A2}}, \neg\mathtt{U}^{\mathtt{B2}}_{\mathtt{A1}}, \neg\mathtt{U}^{\mathtt{B2}}_{\mathtt{A2}}, \neg\mathtt{U}^{\mathtt{C1}}_{\mathtt{B2}}, \neg\mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}}, \neg\mathtt{U}^{\mathtt{C2}}_{\mathtt{B2}}, \neg\mathtt{U}^{\mathtt{C1}}_{\mathtt{A1}}\}$ *as soft constraints. The optimal MaxSAT models are* $\{\mathtt{A1}, \mathtt{B1}, \mathtt{C1}, \mathtt{U}^{\mathtt{C1}}_{\mathtt{A1}}\}$, $\{\mathtt{A1}, \mathtt{B1}, \mathtt{C2}, \mathtt{U}^{\mathtt{C2}}_{\mathtt{B1}}\}$, *and* $\{\mathtt{A2}, \mathtt{B1}, \mathtt{C1}, \mathtt{U}^{\mathtt{B1}}_{\mathtt{A2}}\}$, *giving optimal decoding assignments* $I_1$, $I_2$, *and* $I_3$: $I_1(\mathtt{n}_1) = \mathtt{A1}$, $I_1(\mathtt{n}_2) = \mathtt{B1}$, $I_1(\mathtt{n}_3) = \mathtt{C1}$, $I_2(\mathtt{n}_1) = \mathtt{A1}$, $I_2(\mathtt{n}_2) = \mathtt{B1}$ $I_2(\mathtt{n}_3) = \mathtt{C2}$, $I_3(\mathtt{n}_1) = \mathtt{A2}$, $I_3(\mathtt{n}_2) = \mathtt{B1}$, *and* $I_3(\mathtt{n}_3) = \mathtt{C1}$.

In the next proposition, we show that the partial MaxSAT approach finds the same optimal decoding assignments as the Acc function. For this, we require some subsidiary notation. Let $\mathsf{Const}(M, K)$ be a tuple $C = (H, S)$ where $H$ is a set of hard constraints and $S$ is a set of soft constraints for argument map $M$ and knowledgebase $K$ as defined in Definition 11, and for a tuple of hard and soft constraints $C = (H, S)$, let $\mathsf{Enum}(C)$ be the set of pairs $(X, V)$ where $X$ is a model, represented by a set of literals true in the model, that satisfies the hard constraints $H$ and $V$ is the number of weak constraints in $S$ not satisfied by $X$. Also, let $\mathsf{Trans}(X)$ denote the decoding assignment $I_d$ where for each $n \in N$, $I_d(n) = A$ and $X \cap \mathsf{Cands}(K, I_e(n)) = \{A\}$. So $\mathsf{Trans}(X)$ selects the decoding candidate for the node that appears in the model.

| Number of nodes | 5 | 10 | 15 | 20 |
|---|---|---|---|---|
| $c = 2$ | 0.0013 | 0.143 | 7.261 | timeout |
| $c = 3$ | 0.0102 | 11.85 | timeout | timeout |

Table 1: Results for average time (sec) for finding an optimal decoding assignment for a randomly generated argument map (average of 20 randomly generated argument maps) with $c \in \{2, 3\}$ candidate decodings per node and a timeout of 20 seconds. For c = 3 with 10 nodes, there is one timeout and so average is of 19 maps.

**Proposition 4.** *For argument map $M = (N, P, C, L)$, support relation $R^+$ (resp. attack relation $R^-$), and default knowledgebase $K$, there is $(X, V) \in \mathsf{Enum}(\mathsf{Const}(M, K))$ s.t. for all $(X', V') \in \mathsf{Enum}(\mathsf{Const}(M, K))$, $X \leq X'$ iff there is $I_d \in \arg\max_{I_d \in \mathsf{Assigns}(M,K)} \mathsf{Acc}(R^+, R^-, I_d, N, L)$ s.t. $\mathsf{Trans}(X) = I_d$ and $\mathsf{Acc}(R^+, R^-, I_d, N, L) = (1 - V)/|N|^2$.*

In order to investigate the viability of our approach, we implemented a system (see supplementary material for code) that takes an argument map, and the attack and support relations between a set of decodings, and determines an optimal decoding assignment using the PySAT MaxSAT library [Ignatiev *et al.*, 2018] running on a Windows 11 laptop with an AMD Ryzen 7 5700U processer and 8GB RAM. For evaluation, we randomly generated argument maps with probability of 0.1 of a support or attack between each pair of nodes, and for each pair of nodes, we randomly generated the relationships between candidate decodings with a probability of 0.9 of the decodings agreeing with a support or attack label between the nodes. The results in Figure 8 show the time taken increases exponentially with the number of nodes in the argument map, and the number of decodings per node.

Even for a larger argument map with 100 nodes, and 50 candidates per node, the time is below 3 seconds. Handling 100 nodes is practically useful as human-generated argumentation often involves fewer than 100 arguments (e.g. [Lynch *et al.*, 2012]). Handling 50 decodings per node also seems useful as it would support a large default knowledgebase for decoding. We will investigate this in future work.

Our partial MaxSAT method is significantly better than a naive approach based on constructing and evaluating all the logical assignment functions using Equation 2. We implemented the naive approach in python on the same machine. The results in Table 1 show that the naive approach only works for a much smaller number of candidates and nodes.

## 6 Discussion

In this paper, we have provided a logic-based framework for representing the explicit and implicit aspects of each argument in an argument map. We assume that the text in an argument map can be used to give the initial enthymeme assignment. With the advent of deep learning and large language models, it appears feasible to develop robust, and scalable, methods for translating the natural language text into logic (e.g. [Singh *et al.*, 2020; Levkovskyi and Li, 2021; Lu *et al.*, 2022; Pan *et al.*, 2023; Olausson *et al.*, 2023; Lalwani *et al.*, 2024; Lee *et al.*, 2025]. Therefore, it appears feasible to translate the text that appears in the premises and claims of the nodes in the argument map into formulae of classical logic and therefore assign a full enthymeme to each node. Our proposal allows for the full enthymemes in an initial enthymeme assignment to be formalized, and then systematically decoded using a default knowledgebase. Using default logic as the basis for representing implicit knowledge provides a rich and clear formalism. Optimal decoding assignments can be generated and compared, and differences between a decoding, and the original argument map, can be investigated. So our methods offer automated analysis of the output of argument mining by clarifying the implicitness. This is a first step towards investigating the ambiguity and uncertainty arising with enthymemes in real-world argumentation. Whilst, we have only discussed the propositional version of default logic in this paper, the first-order version of default logic can be used directly with our proposal for richer representation and reasoning with arguments.

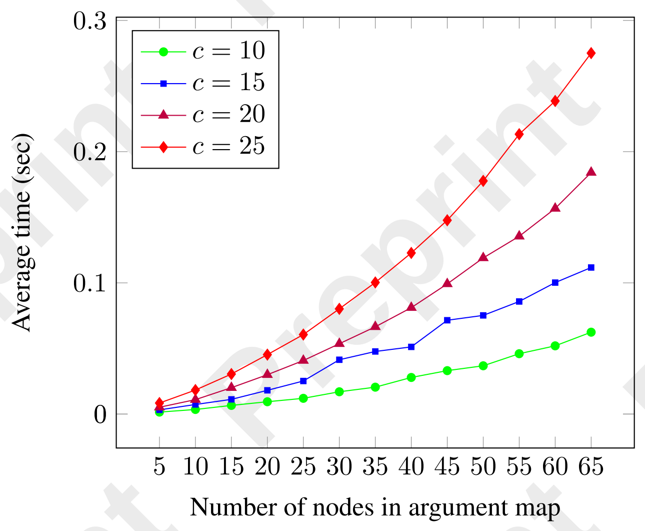


Figure 8: Experimental results for average time taken for finding an optimal decoding assignment for a randomly generated argument map (average of 100 randomly generated argument maps). Each line is for $c \in \{10, 15, 20, 25\}$ candidate decodings per node.

Our approach subsumes deductive argumentation with classical logic since classical formalae are a special case of normal default rules where the precondition is tautology. As a result it is straightforward to capture (and extend upon) a wide range of attack relationships [Besnard and Hunter, 2001; Gorogiannis and Hunter, 2011] and support relationships [Hunter, 2023]. In contrast to ASPIC+ [Modgil and Prakken, 2014], assumption-based argumentation (ABA) [Toni, 2014], and proposals for using default logic for modelling arguments [Prakken, 1993; Santos and Martins, 2008], our proposal clearly demarks the implicit premises and implicit claims in a structured argument representation of an enthymeme, and it goes beyond these proposals by offering a wide range of definitions for support and attack between arguments. More importantly, our proposal introduces the modelling of implicit claims, and how decoding can be optimized in order to maximize the agreement with the labeling of an argument map.

## Acknowledgements

The work by Victor David was supported by the French government, managed by the Agence Nationale de la Recherche under the Plan d'Investissement France 2030, as part of the Initiative d'Excellence d'Université Côte d'Azur under the reference ANR-15-IDEX-01.

## References

[Besnard and Hunter, 2001] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.

[Besnard *et al.*, 2014] Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. Introduction to structured argumentation. *Argumentation and Computation*, 5(1):1–4, 2014.

[Black and Hunter, 2008] Elizabeth Black and Anthony Hunter. Using enthymemes in an inquiry dialogue system. In *Proceedings of AAMAS'08*, pages 437–444. IFAAMAS, 2008.

[Black and Hunter, 2012] Elizabeth Black and Anthony Hunter. A relevance-theoretic framework for constructing and deconstructing enthymemes. *Journal of Logic and Computation.*, 22(1):55–78, 2012.

[Brewka, 1991] Gerhard Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, 1991.

[David and Hunter, 2025] Victor David and Anthony Hunter. Proof and code appendix, 2025. http://www0.cs.ucl.ac.uk/staff/a.hunter/papers/ijcai25.zip.

[Davis, 2017] Ernest Davis. Logical formalizations of commonsense reasoning: A survey. *Journal of Artificial Intelligence Research*, 59:651–723, 2017.

[de Saint-Cyr, 2011a] Florence Dupin de Saint-Cyr. A first attempt to allow enthymemes in persuasion dialogs. In Franck Morvan, A Min Tjoa, and Roland R. Wagner, editors, *Proceedings of DEXA'11*, pages 332–336. IEEE Computer Society, 2011.

[de Saint-Cyr, 2011b] Florence Dupin de Saint-Cyr. Handling enthymemes in time-limited persuasion dialogs. In *Proceedings of SUM'11*, volume 6929 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 2011.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Nicola Leone. Semantics and complexity of abduction from default theories. *Artificial Intelligence*, 90(1-2):177–223, 1997.

[Gorogiannis and Hunter, 2011] Nicos Gorogiannis and Anthony Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9-10):1479–1497, 2011.

[Hosseini *et al.*, 2014] Seyed Ali Hosseini, Sanjay Modgil, and Odinaldo Rodrigues. Enthymeme construction in dialogues using shared knowledge. In *Proc. of COMMA'14*, volume 266 of *FAIA*, pages 325–332. IOS Press, 2014.

[Hunter, 2007] Anthony Hunter. Real arguments are approximate arguments. In *Proceedings of AAAI'07*, pages 66–71. MIT Press, 2007.

[Hunter, 2023] Anthony Hunter. Some options for instantiation of bipolar argument graphs with deductive arguments. *CoRR*, abs/2308.04372, 2023.

[Ignatiev *et al.*, 2018] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*, pages 428–437, 2018.

[Lalwani *et al.*, 2024] Abhinav Lalwani, Lovish Chopra, Christopher Hahn, Caroline Trippel, Zhijing Jin, and Mrinmaya Sachan. NL2FOL: translating natural language to first-order logic for logical fallacy detection. *CoRR*, abs/2405.02318, 2024.

[Lawrence and Reed, 2019] John Lawrence and Chris Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, December 2019.

[Lee *et al.*, 2025] Jinu Lee, Qi Liu, Runzhi Ma, Vincent Han, Ziqi Wang, Heng Ji, and Julia Hockenmaier. Entailment-preserving first-order logic representations in natural language entailment. *CoRR*, abs/2502.16757, 2025.

[Leiva *et al.*, 2023] Diego S. Orbe Leiva, Sebastian Gottifredi, and Alejandro Javier García. Automatic knowledge generation for a persuasion dialogue system with enthymemes. *International Journal of Approximate Reasoning*, 160:108963, 2023.

[Levkovskyi and Li, 2021] Oleksii Levkovskyi and Wei Li. Generating predicate logic expressions from natural language. In *SoutheastCon 2021*, pages 1–8, 2021.

[Lu *et al.*, 2022] Xuantao Lu, Jingping Liu, Zhouhong Gu, Hanwen Tong, Chenhao Xie, Junyang Huang, Yanghua Xiao, and Wenguang Wang. Parsing natural language into propositional and first-order logic with dual reinforcement learning. In *Proceedings of COLING'22*, pages 5419–5431. International Committee on Computational Linguistics, 2022.

[Lynch *et al.*, 2012] Collin F. Lynch, Kevin D. Ashley, and Mohammad Hassan Falakmasir. Comparing argument diagrams. In *Proceedings of JURIX'12*, volume 250 of *Frontiers in Artificial Intelligence and Applications*, pages 81–90. IOS Press, 2012.

[Modgil and Prakken, 2014] Sanjay Modgil and Henry Prakken. The *ASPIC*$^+$ framework for structured argumentation: a tutorial. *Argument and Computation*, 5(1):31–62, 2014.

[Olausson *et al.*, 2023] Theo Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of EMNLP'23*, pages 5153–5176. Association for Computational Linguistics, 2023.

[Pan *et al.*, 2023] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of EMNLP'23*, pages 3806–3824. Association for Computational Linguistics, 2023.

[Panisson *et al.*, 2022] Alison R. Panisson, Peter McBurney, and Rafael H. Bordini. Towards an enthymeme-based communication framework in multi-agent systems. In *Proceedings of KR'22*, 2022.

[Prakken, 1993] Henry Prakken. An argumentation framework in default logic. *Annals of Mathematics and Artificial Intelligence*, 9:93–132, 1993.

[Reiter, 1980] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Santos and Martins, 2008] Emanuel Santos and João Pav̄ao Martins. A default logic based framework for argumentation. In *Proceedings of ECAI'08*, pages 859–860, 2008.

[Singh *et al.*, 2020] Hrituraj Singh, Milan Aggarwal, and Balaji Krishnamurthy. Exploring neural models for parsing natural language into first-order logic. *CoRR*, abs/2002.06544, 2020.

[Toni, 2014] Francesca Toni. A tutorial on assumption-based argumentation. *Argument and Computation*, 5(1):89–117, 2014.

[Walton, 2001] Douglas Walton. Enthymemes, common knowledge and plausible inference. *Philosophy and Rhetoric*, 34(2):93–112, 2001.

[Walton, 2019] Douglas Walton. Argumentation schemes and their application to argument mining. *Windsor Studies in Argumentation*, 8:177–211, 2019.

[Xydis *et al.*, 2020] Andreas Xydis, Christopher Hampson, Sanjay Modgil, and Elizabeth Black. Enthymemes in dialogues. In *Proceedings of COMMA'20*, volume 326 of *FAIA*, pages 395–402. IOS Press, 2020.

[Xydis *et al.*, 2024] Andreas Xydis, Ionut Moraru, and Elizabeth Sklar. Strategising in dialogues handling forward extension of enthymemes. In *Proceedings of COMMA'24*, volume 388 of *FAIA*, pages 337–348. IOS Press, 2024.