Tableaux for Acceptance Logic

Mathijs de Boer[1], Andreas Herzig[2], Tiago de Lima[3], and Emiliano Lorini[2]

[1] University of Luxembourg, Luxembourg
[2] IRIT, University of Toulouse 3, France
[3] Eindhoven University of Technology, The Netherlands

Abstract. We continue the work initiated in [1–3], where the acceptance logic, a modal logic for modelling individual and collective acceptances was introduced. This logic is aimed at capturing the concept of acceptance *qua* member of an institution as the kind of attitude that agents are committed to when they are "functioning as members of an institution". Acceptance logic can also be used to model judgement aggregation: it deals with how a collective acceptance of the members of an institution about a certain fact φ is created from the individual acceptances of the members of the institution. The contribution of this paper is to present a tableau method for the logic of acceptance. The method automatically decides whether a formula of the logic of acceptance is satisfiable thereby providing an automated reasoning procedure for judgement aggregation in the logic of acceptance.

Key words: Semantic tableaux method, acceptance logic, judgement aggregation, discursive dilemma

1 Introduction

The notion of 'acceptance' has been extensively studied in philosophy and social sciences where several authors have distinguished it from the classical notion of belief (see [4–6] for instance). Other authors have been interested in studying the foundational role of acceptance in the existence and in the dynamics of groups and institutions. It has been stressed in [7] (see also [8]) that the existence and the dynamics of an institution depend on the acceptances of the norms and the rules of the institution by the members of the institution. For example, for a certain norm to be a norm of institution x, all members of institution x must accept such norm to be valid. This relationship between acceptance and institutions was already emphasised in the philosophical doctrine of Legal Positivism [9]. According to Hart, the foundations of a normative system or institution consist of adherence to, or acceptance of, an ultimate rule of recognition by which the validity of any rule of the institution may be evaluated.[4]

In some recent works [1–3] we have presented a logical framework in which such relationship between acceptances and institutions can be formally studied.

[4] In Hart's theory, the rule of recognition is the rule which specifies the ultimate criteria of validity in a legal system.

We conceive institutions as rule-governed social practices on the background of which the agents reason. For example, take the case of a game like Cluedo. The institutional context is the rule-governed social practice which the agents conform to in order to be competent players and on the background of which agents reason. In the context of Cluedo, an agent accepts that something has happened *qua* player of Cluedo (e.g., the agent accepts that Mrs. Red is the murderer *qua* player of Cluedo). Our logic is aimed at capturing the state of acceptance *qua* member of an institution as the kind of acceptance one is committed to when one is "functioning as a member of the institution" [7]. Moreover, it enables to formalise the concept of 'collective acceptance' of groups of agents. Following [10, 7], we conceive a collective acceptance held by a set of agents G *qua* members of a certain institution x as the kind of acceptance the agents in G are committed to when they are "functioning together as members of this institution". For example, in the context of Greenpeace agents (collectively) accept that their mission is to protect the Earth *qua* members of Greenpeace. The state of acceptance *qua* members of Greenpeace is the kind of acceptance these agents are committed to when they are functioning together as members of Greenpeace. Thus, in our logical framework a collective acceptance by a set of agents G is based on the identification of the agents in G as members of a certain institution (or group, team, organisation, etc.) and on the fact that the agents in G recognise each other as members of the same institution (or group, team, organisation, etc.).

More recently [11], we have shown that our logic of acceptance can also be applied to modelling some interesting aspects of judgement aggregation. In the logic of acceptance the problem of judgement aggregation is a particular case of the problem of explaining how collective acceptance of the members of a certain group about a certain fact φ is created from the individual acceptances of the members of this group.

The contribution of this article is to present a tableau method for the logic of acceptance we introduced in [1, 2]. The method automatically decides whether a formula of the logic of acceptance is satisfiable thereby providing an automated reasoning procedure for making judgement aggregation in modal logic.

The remainder of the paper is organised as follows. First, in Section 2 we briefly present acceptance logic. Then, in Section 3 we present our tableau method. In Section 4 we apply it to a classical scenario in judgment aggregation, the so-called Discursive Dilemma [12, 13]. And finally, Section 5 concludes.

2 Acceptance logic

The logic AL (*Acceptance Logic*) was introduced in [1, 2]. It allows to express that some agents identify themselves as members of a certain institution and what (groups of) agents accept while functioning together as members of an institution. The principles of AL clarify the relationships between individual acceptance (acceptances of individual agents) and collective acceptance (acceptances of groups of agents).

2.1 Syntax

Assume a finite non-empty set X of labels denoting institutional contexts, a finite non-empty set N of labels denoting agents and a countable set P of atomic formulae. We use $2^{N\star}$ to denote the set $2^N \setminus \emptyset$.

The language $\mathcal{L}_{\mathrm{AL}}$ of acceptance logic is the set of formulae φ defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathrm{A}_{G:x}\varphi$$

where p ranges over P, G ranges over $2^{N\star}$, and x ranges over X.

The other classical Boolean connectives $\wedge$, $\rightarrow$, $\leftrightarrow$, $\top$ (tautology) and $\bot$ (contradiction) are defined using $\vee$ and $\neg$ in the usual manner. And for simplicity, we write $i{:}x$ instead of $\{i\}{:}x$.

The formula $\mathrm{A}_{G:x}\varphi$ reads 'the agents in G accept that φ while functioning together as members of institution x'. For example, $\mathrm{A}_{G:\mathit{Greenpeace}}\mathit{protectEarth}$ expresses that the agents in G accept that the mission of Greenpeace is to protect the Earth while functioning as activists in the context of Greenpeace; and $\mathrm{A}_{i:\mathit{Catholic}}\mathit{PopeInfallible}$ expresses that agent i accepts that the Pope is infallible while functioning as a Catholic in the context of the Catholic Church.

The same agent may accept contradictory propositions in two different contexts. For example, while functioning as a Catholic, agent i accepts that killing is forbidden, and while functioning as a soldier i accepts that killing is allowed.

The formula $\mathrm{A}_{G:x}\bot$ has to be read 'agents in G are not functioning together as members of institution x'. This means that we assume that functioning as a member of an institution is, at least in this minimal sense, a rational activity. Conversely, $\neg\mathrm{A}_{G:x}\bot$ has to be read 'agents in G are functioning together as members of institution x'. Thus, $\neg\mathrm{A}_{G:x}\bot \wedge \mathrm{A}_{G:x}\varphi$ stands for 'agents in G are functioning together as members of institution x and they accept that φ while functioning together as members of x' or simply 'agents in G accept that φ *qua* members of institution x'. This is a case of group acceptance. For the individual case, the formula $\neg\mathrm{A}_{i:x}\bot \wedge \mathrm{A}_{i:x}\varphi$ has to be read 'agent i accepts that φ *qua* member of institution x'.

2.2 Semantics and axiomatisation

We use a standard possible worlds semantics. Let the set of all pairs of non-empty sets of agents and institutional contexts be $\Delta = \{G{:}x \mid G \in 2^{N\star} \text{ and } x \in X\}$. An acceptance model is a triple $\langle W, \mathcal{A}, \mathcal{V}\rangle$ where: W is a non-empty set of possible worlds, $\mathcal{A} : \Delta \rightarrow W \times W$ maps every $G{:}x \in \Delta$ to a relation $\mathcal{A}(G{:}x)$ between possible worlds in W and $\mathcal{V} : P \rightarrow 2^W$ is valuation function associating a set of possible worlds $\mathcal{V}(p) \subseteq W$ to each atomic formula p of P.

Instead of $\mathcal{A}(G{:}x)$ we write $\mathcal{A}_{G:x}$, and we use $\mathcal{A}_{G:x}(w)$ to denote the set $\{w' \mid \langle w, w'\rangle \in \mathcal{A}_{G:x}\}$. $\mathcal{A}_{G:x}(w)$ is the set of worlds that is acceptable by the agents in G while functioning together as members of institution x.

Given $M = \langle W, \mathcal{A}, \mathcal{V}\rangle$ and $w \in W$, the pair $\langle M, w\rangle$ is a pointed acceptance model. The satisfaction relation $\models$ between formulae of $\mathcal{L}_{\mathrm{AL}}$ and pointed acceptance models $\langle M, w\rangle$ is defined as usual for atomic propositions, negation and disjunction. The satisfaction relation for acceptance operators is the following:

$$M, w \models \mathrm{A}_{G:x}\varphi \quad \text{iff} \quad M, w' \models \varphi \text{ for all } w' \in \mathcal{A}_{G:x}(w)$$

Validity of a formula φ (noted: $\models \varphi$) is defined as usual.

The axiomatisation of AL is presented in Figure 1. As usual, the K-principles are the axioms and inference rules of the basic modal logic K.

(K)	All K-principles for the operators $\mathrm{A}_{G:x}$	
(4∗)	$\mathrm{A}_{G:x}\varphi \rightarrow \mathrm{A}_{H:y}\mathrm{A}_{G:x}\varphi$	(if $H \subseteq G$)
(5∗)	$\neg\mathrm{A}_{G:x}\varphi \rightarrow \mathrm{A}_{H:y}\neg\mathrm{A}_{G:x}\varphi$	(if $H \subseteq G$)
(Inc)	$(\neg\mathrm{A}_{G:x}\bot \wedge \mathrm{A}_{G:x}\varphi) \rightarrow \mathrm{A}_{H:x}\varphi$	(if $H \subseteq G$)
(Una)	$\mathrm{A}_{G:x}(\bigwedge_{i \in G} \mathrm{A}_{i:x}\varphi \rightarrow \varphi)$	

Fig. 1. Axiomatisation of acceptance logic.

Axioms **4∗** and **5∗** are introspection axioms: when the agents in a set G function together as members of institution x then, for all $y \in X$ and all $H \subseteq G$, the agents in H have access to all the facts that are accepted (or that are not accepted) by the agents in G. In particular, if the agents in G (do not) accept that φ while functioning together as members of institution x then, while functioning together as members of institution x, the agents of every subset H of G accept that agents in G (do not) accept that φ.

Axiom **Inc** says that, if the agents in G accept that φ *qua* members of institution x then every subset H of G accepts φ while functioning together as members of institution x. This means that what is accepted by the agents in G *qua* members of institution x are necessarily accepted by agents in all of G's subsets with respect to the same institutional context x. Axiom **Inc** describes the *top down* process leading from G's collective acceptance to the individual acceptances of G's members.

Axiom **Una** expresses a unanimity principle according to which the agents in G, while functioning together as members of institution x, accept that if each of them individually accepts that φ while functioning as member of x, then φ is the case. This axiom describes the *bottom up* process leading from individual acceptances of the members of G to the collective acceptance of the group G.

In order to make our axioms valid we impose the following constraints on acceptance models, for any world $w \in W$, institutional context $x \in X$, and groups $G, H \in 2^{N^\star}$ such that $H \subseteq G$:

(**C**.**4∗**) if $w_2 \in \mathcal{A}_{H:y}(w_1)$ and $w_3 \in \mathcal{A}_{G:x}(w_2)$ then $w_3 \in \mathcal{A}_{G:x}(w_1)$;

(**C**.**5∗**) if $w_2 \in \mathcal{A}_{H:y}(w_1)$ and $w_3 \in \mathcal{A}_{G:x}(w_1)$ then $w_3 \in \mathcal{A}_{G:x}(w_2)$;

(**C**.**Inc**) if $\mathcal{A}_{G:x}(w) \neq \emptyset$ then $\mathcal{A}_{H:x}(w) \subseteq \mathcal{A}_{G:x}(w)$;

(**C**.**Una**) if $w_2 \in \mathcal{A}_{G:x}(w_1)$ then $w_2 \in \bigcup_{i \in G} \mathcal{A}_{i:x}(w_2)$.

Axiom **4**$*$ corresponds to semantic constraint **C.4**$*$, Axiom **5**$*$ corresponds to **C.5**$*$, Axiom **Inc** to **C.Inc**, and **Una** to **C.Una** (in the sense of correspondence theory). We also note that **C.4**$*$ and **C.5**$*$ together are equivalent to the following semantic constraint: if $w_2 \in \mathcal{A}_{H:y}(w_1)$ then $\mathcal{A}_{G:x}(w_1) = \mathcal{A}_{G:x}(w_2)$. Thus, the acceptance models considered here are exactly the same as proposed in [1–3]. The theorem below has been shown in [2].

Theorem 1. *The axiomatisation in Figure 1 is sound and complete with respect to the class of acceptance models satisfying constraints* **C.4**$*$*,* **C.5**$*$*,* **C.Inc** *and* **C.Una**.

2.3 Discussion about the monotonicity principle

One may think that the following principle would be desirable in Acceptance Logic:

$$\textbf{(Mon)} \quad \neg \mathrm{A}_{G:x}\bot \rightarrow \neg \mathrm{A}_{H:x}\bot \quad (\text{if } H \subseteq G)$$

which corresponds to semantic constraint: if $\mathcal{A}_{G:x}(w) \neq \emptyset$ then $\mathcal{A}_{H:x}(w) \neq \emptyset$. Principle **Mon** expresses a property of monotonicity about institution membership. It was also discussed from a different perspective in our previous works on Acceptance Logic [1, 2].

We prefer not including **Mon** in the current version of AL because we are interested in strong notions of 'constituted group' and 'group identification' which are formally expressed by constructions $\neg \mathrm{A}_{G:x}\bot$. As we said above, $\neg \mathrm{A}_{G:x}\bot$ means "the agents in G are functioning together as members of the institution x" or, stated differently, "G constitutes a group of members of the institution x". We suppose here that the latter sentences just express that: every agent in G identifies himself as a member of institution x and recognizes G as a *group of members* of institution x. Under this assumption, **Mon** is not valid. The following example illustrates this point. Imagine that the eleven agents in $\{1, 2, \ldots, 11\}$ constitute a football team (i.e. $\neg \mathrm{A}_{\{1,2,\ldots,11\}:team}\bot$). This means that every agent in $\{1, 2, \ldots, 11\}$ identifies himself as a member of the football team and recognizes $\{1, 2, \ldots, 11\}$ as a football team. This does not entail that $\{1, 2, \ldots, 10\}$ constitute a football team (i.e. $\neg \mathrm{A}_{\{1,2,\ldots,10\}:team}\bot$). Indeed, it is not the case that every agent in $\{1, 2, \ldots, 10\}$ recognizes $\{1, 2, \ldots, 10\}$ as a football team. (Only ten players do not constitute a football team!).

It worth noting that **Mon** becomes a reasonable principle under a different reading of the construction $\neg \mathrm{A}_{G:x}\bot$. Namely, suppose that $\neg \mathrm{A}_{G:x}\bot$ just means: every agent in G identifies himself as a member of institution x and recognises every agent in the set of agents G as a member of institution x. Under this assumption, $\neg \mathrm{A}_{G:x}\bot$ should imply $\neg \mathrm{A}_{H:x}\bot$, for $H \subseteq G$.

3 The tableau method

In this section we present a proof method for AL that uses semantic tableaux. As a typical tableaux method, given a formula φ, it systematically tries to construct a model for it. When it fails, φ is inconsistent and thus, its negation is valid.

Each formula in the tableau is prefixed by a natural number that stands for a possible world in the model under construction, similar to the notion used by Fitting ([14, Chapter 8]).

Definition 1 (Labelled formula). *A* labelled formula *is a pair of the form* $\langle n, \varphi\rangle$ *such that* $n \in \mathbb{N}$ *and* $\varphi \in \mathcal{L}_{\mathrm{AL}}$*.*

Our method also builds the relations $\mathcal{A}_{G:x}$ between possible worlds that form the model. These relations are represented in tableau by means of arrows between possible worlds, which are represented as triples of the form $\langle G{:}x, n, n'\rangle$. That is, the tableau contains a set of labelled formulae and also a set of such triples, as defined in the sequel.

Definition 2 (Branch). *A* branch *is a pair of the form* $\langle L, S\rangle$ *such that* L *is a set of labelled formulae and* $S \subseteq (\Delta \times \mathbb{N} \times \mathbb{N})$*.*

Definition 3 (Tableau). *Let* $\varphi \in \mathcal{L}_{\mathrm{AL}}$*. A tableau for* φ *is a set of branches* T *inductively defined as follows:*

- $T = \{\langle\{\langle 0, \varphi\rangle\}, \emptyset\rangle\}$*. This is called the* initial tableau for φ*.*
- $T' = (T \setminus \{\langle L, S\rangle\}) \cup \{\langle L'_1, S'_1\rangle, \langle L'_2, S'_2\rangle, \ldots, \langle L'_n, S'_n\rangle\}$*, where* T *is a tableau for* φ *containing the branch* $\langle L, S\rangle$ *and each* $\langle L'_i, S'_i\rangle$ *is a branch generated by one of the* tableau rules *defined below: (A more standard presentation of some of these tableau rules is given in Figure 2.)*
  - **(R.¬)** *If* $\langle n, \neg\neg\varphi\rangle \in L$ *then generate* $L'_1 = L \cup \{\langle n, \varphi\rangle\}$ *and* $S'_1 = S$*.*
  - **(R.∧)** *If* $\langle n, \varphi_1 \wedge \varphi_2\rangle \in L$ *then generate* $L'_1 = L \cup \{\langle n, \varphi_1\rangle, \langle n, \varphi_2\rangle\}$ *and* $S'_1 = S$*.*
  - **(R.∨)** *If* $\langle n, \neg(\varphi_1 \wedge \varphi_2)\rangle \in L$ *then generate* $L'_1 = L \cup \{\langle n, \neg\varphi_1\rangle\}$ *and* $S'_1 = S$*; and also* $L'_2 = L \cup \{\langle n, \neg\varphi_2\rangle\}$ *and* $S'_2 = S$*.*
  - **(R.□)** *if* $\langle n, \mathrm{A}_{G:x}\varphi\rangle \in L$ *and* $\langle G{:}x, n, n'\rangle \in S$ *then generate* $L'_1 = L \cup \{\langle n', \varphi\rangle\}$ *and* $S'_1 = S$*.*
  - **(R.◇)** *if* $\langle n, \neg\mathrm{A}_{G:x}\varphi\rangle \in L$ *then generate* $L'_1 = L \cup \{\langle n', \neg\varphi\rangle\}$ *and* $S'_1 = S \cup \{\langle G{:}x, n, n'\rangle\}$*, for some* n' *that does not occur in* L*.*
  - **(R.4∗)** *if* $H \subseteq G$ *and* $\langle H{:}y, n, n'\rangle$*,* $\langle G{:}x, n', n''\rangle \in S$ *then generate* $L'_1 = L$ *and* $S'_1 = S \cup \{\langle G{:}x, n, n''\rangle\}$*.*
  - **(R.5∗)** *If* $H \subseteq G$ *and* $\langle H{:}y, n, n'\rangle, \langle G{:}x, n, n''\rangle \in S$ *then generate* $L'_1 = L$ *and* $S'_1 = S \cup \{\langle G{:}x, n', n''\rangle\}$*.*
  - **(R.Inc)** *If* $H \subseteq G$ *and* $\langle H{:}x, n, n'\rangle \in S$ *then generate* $L'_1 = L$ *and* $S'_1 = S \cup \{\langle G{:}x, n, n'\rangle\}$*; and also* $L'_2 = L \cup \{\langle n, \mathrm{A}_{G:x}\bot\rangle\}$ *and* $S'_2 = S$*.*
  - **(R.Una)** *If* $G = \{i_1, \ldots, i_k\}$ *and* $\langle G{:}x, n, n'\rangle \in S$ *then generate* $L'_1 = L$ *and* $S'_1 = S \cup \{\langle \{i_1\}{:}x, n', n'\rangle\}$*;* $L'_2 = L$ *and* $S'_2 = S \cup \{\langle \{i_2\}{:}x, n', n'\rangle\}$*; …;* $L'_k = L$ *and* $S'_k = S \cup \{\langle \{i_k\}{:}x, n', n'\rangle\}$*.*

Rules **R.**¬, **R.**∧ and **R.**∨ work exactly as for classical propositional logic. Rules **R.**□ and **R.**◇ work exactly as for modal logic K, as proposed, e.g., in [14]. The other rules are meant to ensure that the model to be created will be an acceptance model. Tableau Rule **R.4∗** implements semantic constraint **C.4∗**, Tableau Rule **R.5∗** implements constraint **C.5∗**, Rule **R.Inc** implements **C.Inc** and **R.Una** implements **C.Una**.

$$(\mathbf{R}.\Box) \qquad \frac{\langle n, \mathrm{A}_{G:x}\varphi\rangle; \langle G{:}x, n, n'\rangle}{\langle n', \varphi\rangle;}$$

$$(\mathbf{R}.\Diamond) \qquad \frac{\langle n, \neg\mathrm{A}_{G:x}\varphi\rangle;}{\langle n', \neg\varphi\rangle; \langle G{:}x, n, n'\rangle} \qquad \text{for a new } n'$$

$$(\mathbf{R}.\mathbf{4*}) \qquad \frac{; \langle H{:}y, n, n'\rangle, \langle G{:}x, n', n''\rangle}{; \langle G{:}x, n, n''\rangle} \qquad \text{where } H \subseteq G$$

$$(\mathbf{R}.\mathbf{5*}) \qquad \frac{; \langle H{:}y, n, n'\rangle, \langle G{:}x, n, n''\rangle}{; \langle G{:}x, n', n''\rangle} \qquad \text{where } H \subseteq G$$

$$(\mathbf{R}.\mathbf{Inc}) \qquad \frac{; \langle H{:}x, n, n'\rangle}{; \langle G{:}x, n, n'\rangle \,\big|\, \langle n, \mathrm{A}_{G:x}\bot\rangle;} \qquad \text{where } H \subseteq G$$

$$(\mathbf{R}.\mathbf{Una}) \qquad \frac{; \langle G{:}x, n, n'\rangle}{; \langle \{i_1\}{:}x, n', n'\rangle \big| \dots \big| ; \langle \{i_k\}{:}x, n', n'\rangle} \qquad \text{where } G = \{i_1, \dots, i_k\}$$

Fig. 2. Tableau rules

Definition 4 (Closed tableau). *The set of labelled formulae L is* closed *if and only if $\{\langle n, \varphi\rangle, \langle n, \neg\varphi\rangle\} \subseteq L$, for some n and φ. A branch is closed if and only if its set of labelled formulae is closed. A tableau is closed if and only if all its branches are closed. A tableau is* open *if and only if it is not closed.*

Example 1. Now, let us see how the method can be used to show that the formula: $(\mathrm{A}_{ij:x}\mathrm{A}_{i:x}p \wedge \mathrm{A}_{ij:x}\mathrm{A}_{j:x}p) \rightarrow \mathrm{A}_{ij:x}p$ is valid in acceptance logic. As we will see, if there is a closed tableau for φ, then no model satisfies φ, which means that $\neg\varphi$ is valid. Therefore, if we provide a closed tableau for the negation of our formula above, we show its validity.

Such a tableau is given in Figure 3. Each line of the figure displays either a labelled formula, an arrow or both, which are the elements of the tableau branches. The number in parentheses on the left is used to identify the line. On the right, also in parentheses, we find the rule that generated that line, and what lines has been used in the application of such rule. For example, the labelled formula and arrow in line 6 have been generated by the application of $\mathbf{R}.\Diamond$, using the labelled formula in line 3. Also, lines 9 and 11 have been generated by the application of $\mathbf{R}.\mathbf{Una}$, using the arrow in line 6.

The input formula is in line 1 (we spelt out the abbreviations). The construction of the closed tableau started with the initial tableau for the input formula. The latter corresponds to line 1 alone. Then, a new tableau has been generated by the application of $\mathbf{R}.\wedge$ using the labelled formula in line 1. The latter corresponds to lines 1 and 2 together, and so on. When $\mathbf{R}.\mathbf{Una}$ has been applied using the arrow in line 6, it generated two branches. This is represented in the figure by the vertical line dividing the tableau in two parts after line 8.

<table>
<tr><td colspan="4">(1) $0, (\mathrm{A}_{ij:x}\mathrm{A}_{i:x}p \wedge \mathrm{A}_{ij:x}\mathrm{A}_{j:x}p) \wedge \neg\mathrm{A}_{ij:x}p$</td><td></td><td></td></tr>
<tr><td colspan="4">(2) $0, \mathrm{A}_{ij:x}\mathrm{A}_{i:x}p \wedge \mathrm{A}_{ij:x}\mathrm{A}_{j:x}p$</td><td></td><td>$(\mathbf{R}.\wedge : 1)$</td></tr>
<tr><td colspan="4">(3) $0, \neg\mathrm{A}_{ij:x}p$</td><td></td><td>$(\mathbf{R}.\wedge : 1)$</td></tr>
<tr><td colspan="4">(4) $0, \mathrm{A}_{ij:x}\mathrm{A}_{i:x}p$</td><td></td><td>$(\mathbf{R}.\wedge : 2)$</td></tr>
<tr><td colspan="4">(5) $0, \mathrm{A}_{ij:x}\mathrm{A}_{j:x}p$</td><td></td><td>$(\mathbf{R}.\wedge : 2)$</td></tr>
<tr><td colspan="4">(6) $1, \neg p$</td><td>$ij{:}x, 0, 1$</td><td>$(\mathbf{R}.\Diamond : 3)$</td></tr>
<tr><td colspan="4">(7) $1, \mathrm{A}_{i:x}p$</td><td></td><td>$(\mathbf{R}.\Box : 4, 6)$</td></tr>
<tr><td colspan="4">(8) $1, \mathrm{A}_{j:x}p$</td><td></td><td>$(\mathbf{R}.\Box : 5, 6)$</td></tr>
<tr><td>(9)</td><td>$i{:}x, 1, 1$</td><td>$(\mathbf{R}.\mathbf{Una} : 6)$</td><td>(11)</td><td>$j{:}x, 1, 1$</td><td>$(\mathbf{R}.\mathbf{Una} : 6)$</td></tr>
<tr><td>(10) $1, p$</td><td></td><td>$(\mathbf{R}.\Box : 7, 9)$</td><td>(12) $1, p$</td><td></td><td>$(\mathbf{R}.\Box : 8, 11)$</td></tr>
<tr><td>closed</td><td></td><td>$(6, 10)$</td><td>closed</td><td></td><td>$(6, 12)$</td></tr>
</table>

Fig. 3. A tableau for Example 1

Example 2. On the other hand, if no closed tableau for φ exists, then φ is satisfiable. Let us see what happens when we try to generate a closed tableau for the formula: $\neg\mathrm{A}_{i:x}p \wedge \mathrm{A}_{ij:x}\neg\mathrm{A}_{i:x}p$ which is satisfiable. This is done in Figure 4. Note that one of the branches is closed. On the other hand, no rule can be applied in the rightmost branch, which means that this tableau will remain open. On the leftmost branch we have a rather different phenomenon. We can continue applying the same set of rules indefinitely. This will generate more branches that can be closed, but we can never close all of them. This means that we can also consider such a branch as an open one.

We proceed by proving soundness of the method. But first we need yet another definition.

Definition 5 (Satisfiable branch). *The branch $b = \langle L, S\rangle$ is* satisfiable *if and only if there exists an acceptance model $M = \langle W, \mathcal{A}, \mathcal{V}\rangle$ and a function $f : \mathbb{N} \to W$ such that:*

1. *$\langle f(n), f(n')\rangle \in \mathcal{A}_{G:x}$, for all $\langle G{:}x, n, n'\rangle \in S$; and*
2. *$M, f(n) \models \varphi$, for all $\langle n, \varphi\rangle \in L$.*

Theorem 2 (Soundness). *If there is a closed tableau for $\neg\varphi$ then φ is valid.*

Proof. We show that if φ is satisfiable then there is no closed tableau for φ. It is enough to show that all tableau rules preserve satisfiability. That is, it is enough to show that: if the branch $b = \langle L, S\rangle$ is satisfiable then the set of branches $B = \{b'_1, \ldots, b'_k\}$ generated by any tableau rule contains a satisfiable branch.

<table>
<tr><td colspan="3">(1) $0, \neg \mathrm{A}_{i:x}p \wedge \mathrm{A}_{ij:x}\neg \mathrm{A}_{i:x}p$
(2) $0, \neg \mathrm{A}_{i:x}p$ ($\mathbf{R}.\wedge : 1$)
(3) $0, \mathrm{A}_{ij:x}\neg \mathrm{A}_{i:x}p$ ($\mathbf{R}.\wedge : 1$)
(4) $1, \neg p$ $i{:}x, 0, 1$ ($\mathbf{R}.\Diamond : 2$)</td></tr>
<tr><td colspan="2">(5) $ij{:}x, 0, 1$ ($\mathbf{R}.\mathbf{Inc} : 4$)
(6) $1, \neg \mathrm{A}_{i:x}p$ ($\mathbf{R}.\Box : 3, 5$)
(7) $ij{:}x, 1, 1$ ($\mathbf{R}.\mathbf{5*} : 5$)
(8) $2, \neg p$ $i{:}x, 1, 2$ ($\mathbf{R}.\Diamond : 6$)</td><td rowspan="2">(16) $0, \mathrm{A}_{ij:x}\bot$ ($\mathbf{R}.\mathbf{Inc} : 4$)</td></tr>
<tr><td>(9) $ij{:}x, 1, 2$ ($\mathbf{R}.\mathbf{Inc} : 8$)
(10) $ij{:}x, 0, 2$ ($\mathbf{R}.\mathbf{4*} : 4, 9$)
(11) $2, \neg \mathrm{A}_{i:x}p$ ($\mathbf{R}.\Box : 3, 10$)
(12) $ij{:}x, 2, 2$ ($\mathbf{R}.\mathbf{5*} : 9$)
(13) $3, \neg p$ $i{:}x, 2, 3$ ($\mathbf{R}.\Diamond : 11$)
⋮</td><td>(14) $1, \mathrm{A}_{ij:x}\bot$ ($\mathbf{R}.\mathbf{Inc} : 8$)
(15) $1, \bot$ ($\mathbf{R}.\Box : 14, 7$)
closed (15)</td></tr>
</table>

Fig. 4. A tableau for Example 2

Indeed, and to see that it is enough, suppose that b'_i is satisfiable and closed. Then L'_i contains two labelled formulae $\langle n, \varphi \rangle$ and $\langle n, \neg\varphi \rangle$. Because b'_i is satisfiable, there exists an acceptance model M and a function f such that $M, f(n) \models \varphi$ and $M, f(n) \models \neg\varphi$, which is a contradiction.

Now, suppose that the branch b is satisfiable. The proof that the rules $\mathbf{R}.\neg$ and $\mathbf{R}.\wedge$ preserve satisfiability is straightforward and thus, left as an exercise to the reader. We proceed by showing that the modal rules preserve satisfiability.

$\mathbf{R}.\Box$: $M, f(n) \models \mathrm{A}_{G:x}\varphi$ and $\langle f(n), f(n') \rangle \in \mathcal{A}_{G:x}$ (by hypothesis). Then $M, f(n') \models \varphi$ (by definition).

$\mathbf{R}.\Diamond$: $M, f(n) \models \neg\mathrm{A}_{G:x}\varphi$ (by hypothesis). Then there exists $w' \in \mathcal{A}_{G:x}(f(n))$ such that $M, w' \models \neg\varphi$ (again, by definition). Now, consider the function $f' : \mathbb{N} \to W$ such that $f'(n) = f(n)$, for all n occurring in L, and $f'(n') = w'$. Then $M, f'(n'') \models \varphi''$, for all $\langle n'', \varphi'' \rangle \in L$ (because n' does not occur in L), and $M, f'(n') \models \neg\varphi$.

$\mathbf{R}.\mathbf{4*}$: Let $H \subseteq G$. $\langle f(n), f(n') \rangle \in \mathcal{A}_{H:y}$ and $\langle f(n'), f(n'') \rangle \in \mathcal{A}_{G:x}$ (by hypothesis). Then $\langle f(n), f(n'') \rangle \in \mathcal{A}_{G:x}$, since M is an acceptance model respecting $\mathbf{C}.\mathbf{4*}$. Then, the branch $\langle L'_1, S'_1 \rangle$ is satisfiable.

$\mathbf{R}.\mathbf{5*}$: Let $H \subseteq G$. $\langle f(n), f(n') \rangle \in \mathcal{A}_{H:x}$ and $\langle f(n), f(n'') \rangle \in \mathcal{A}_{G:x}$ (by hypothesis). Then, $\langle f(n'), f(n'') \rangle \in \mathcal{A}_{G:x}$, since M is an acceptance model respecting $\mathbf{C}.\mathbf{5*}$. Then, the branch $\langle L'_1, S'_1 \rangle$ is satisfiable.

$\mathbf{R}.\mathbf{Inc}$: Let $H \subseteq G$. $M, f(n) \models \mathrm{A}_{G:x}\varphi$ and $\langle f(n), f(n') \rangle \in \mathcal{A}_{H:x}$ (by hypothesis). Note that $M, f(n') \not\models \bot$ (because we assume that the branch is satisfiable) then $M, f(n) \models \neg\mathrm{A}_{H:x}\bot$. The latter implies $\langle f(n), f(n') \rangle \in \mathcal{A}_{G:x}$ or $\mathcal{A}_{G:x}(f(n)) = \emptyset$, since M is an acceptance model respecting $\mathbf{C}.\mathbf{Inc}$. Therefore, one of the branches generated by $\mathbf{R}.\mathbf{Inc}$ is satisfiable.

$\mathbf{R}.\mathbf{Una}$: Let $G = \{i_1, \ldots, i_k\}$. $\langle f(n), f(n') \rangle \in \mathcal{A}_{G:x}$ (by hypothesis). Then, $\langle f(n'), f(n') \rangle \in \mathcal{A}_{i_j:x}$ for some $1 \leq j \leq k$, since M is an acceptance model respecting **C**.**Una**. Therefore, one of the branches generated by **R**.**Una** is satisfiable. □

In the sequel we prove completeness. First though, we need another auxiliary definition.

Definition 6 (Saturated tableau). *Let T be a tableau for φ containing the branch $b = \langle L, S\rangle$. The branch b is 'saturated under the tableau rule ρ' if and only if L contains L'_i and S contains S'_i for some branch $\langle L'_i, S'_i\rangle$ generated by the application of the rule ρ to b. A branch is (simply) 'saturated' if and only if it is saturated under all tableau rules. A tableau is saturated if and only if all its branches are saturated.*

Theorem 3 (Completeness). *If φ is valid then there exists a closed tableau for $\neg\varphi$.*

Proof. Let $\langle L, S\rangle$ be an open and saturated branch of the tableau. We build a model $M = \langle W, \mathcal{A}, \mathcal{V}\rangle$ such that $W = \{n \mid \langle n, \varphi\rangle \in L\}$, for some φ; $\mathcal{A}_{G:x}(n) = \{n' \mid \langle G{:}x, n, n'\rangle \in S\}$; and $\mathcal{V}(p) = \{n \mid \langle n, p\rangle \in L\}$.

Clearly, M is an acceptance model, because the branch is saturated under **R**.**4**$*$, **R**.**5**$*$, **R**.**Inc** and **R**.**Una**.

Now, we show that for all $\langle n, \varphi\rangle \in L$, $M, n \models \varphi$. It is done by induction on the structure of φ.

There are two cases in the induction base. (1) $\varphi = p$, i.e., $\langle n, p\rangle \in L$. Then $n \in \mathcal{V}(p)$, iff $M, n \models p$ (by definition). (2) $\varphi = \neg p$, i.e., $\langle n, \neg p\rangle \in L$. Then $\langle n, p\rangle \notin L$, because L is open. Then $n \notin \mathcal{V}(p)$, iff $M, n \models \neg p$ (by definition).

There are five cases in the induction step. (1) $\varphi = \neg\neg\varphi_1$, i.e., $\langle n, \neg\neg\varphi_1\rangle \in L$. Then $\langle n, \varphi_1\rangle \in L$, because L is saturated under **R**.$\neg$. Then $M, n \models \varphi_1$, by induction hypothesis, iff $M, n \models \neg\neg\varphi_1$, by definition. Cases (2) $\varphi = \varphi_1 \wedge \varphi_2$ and (3) $\varphi = \neg(\varphi_1 \wedge \varphi_2)$, are shown analogously as case (1) using rules **R**.$\wedge$ and **R**.$\vee$, respectively. They are left as an exercise to the rader. (4) $\langle n, \mathrm{A}_{G:x}\varphi\rangle \in L$. Then for all $n' \in W$, if $\langle G{:}x, n, n'\rangle \in S$, then $\langle n', \varphi\rangle \in L$ (because L is saturated under rule **R**.$\Box$). Then for all $n' \in W$, if $n' \in \mathcal{A}_{G:x}(n)$, then $M, n' \models \varphi$, by induction hypothesis. Therefore, $M, n \models \mathrm{A}_{G:x}\varphi$. (5) $\langle n, \neg\mathrm{A}_{G:x}\varphi\rangle \in L$. Then there is $n' \in W$ such that $\langle G{:}x, n, n'\rangle \in S$ and $\langle n', \neg\varphi\rangle \in L$ (because L is saturated under rule **R**.$\Diamond$). Then there is $n' \in W$ such that $n' \in \mathcal{A}_{G:x}(n)$ and $M, n' \models \neg\varphi$, by induction hypothesis. Therefore, $M, n \models \neg\mathrm{A}_{G:x}\varphi$. □

Theorem 4 (Termination). *There exists an implementation of the tableau method that halts for every input formula φ.*

Proof (Sketch). We assume an implementation of the tableau method that employs a 'loop-test'. That is, a procedure that, once the latest generated tableau is saturated under all rules but **R**.$\Diamond$, verifies whether the application of the latter rule to the witness formula ψ will generate a world such that the set of labelled formulae having this world as label and the set of arrows involving this world will be included in the respective sets for a different world already present in the branch. If it is the case, then ψ is marked and $\mathbf{R}.\Diamond$ will not be applied using this formula any more.

Then, the argument for termination goes as for logic S4 (as used, e.g., in [15, 16]): the formulae generated by the rules are in a finite set S, and therefore only a finite number of different nodes can be generated by the tableaux procedure. The only difference is that here the finite set S is not just the set of sub-formulae of the initial formula φ as for S4, but its closure, which is the set of sub-formulae of φ union the set of $\mathrm{A}_{G:x}\bot$ such that G and x occur in φ (due to rule **R.Inc**). Moreover, the set of labels of every arrow is finite. □

4 An example: The discursive dilemma

In the recent years many researchers in philosophy, computer science and political sciences have been working on the issue of judgement aggregation (e.g., [17–22]). The problem is: How can a group of individuals aggregate the group members' individual judgements on some interconnected propositions into corresponding collective judgements on these propositions? Such problems occur in different social and legal contexts like committees, legislatures, judiciaries and expert panels.

Our logic of acceptance is a formal framework in which some important aspects of judgement aggregation can be modelled. Moreover, the tableau method for the logic of acceptance presented in Section 3 provides an interesting solution for making automated reasoning about judgement aggregation.

In the logic of acceptance the problem of judgement aggregation is a particular case of the problem of explaining how collective acceptance of the members of a certain group in an institutional context x about a certain fact φ is created from the individual acceptances in x of the members of the same group.

We here consider a well-known problem in judgement aggregation called 'doctrinal paradox' or 'discursive dilemma' [12, 13]. The scenario of the discursive dilemma is a three-member court which has to judge whether a defendant is liable for a breach of contract. According to the legal doctrine, the defendant is liable (*lia*) if and only if he did a certain action (*act*) and he had a contractual obligation not to do this action (*obl*). This is expressed in propositional logic by the connection rule $lia \leftrightarrow (act \wedge obl)$. The three judges use majority rule to decide on this issue. The opinions of the three judges are given in Table 1.

	act	*obl*	$lia \leftrightarrow (act \wedge obl)$	*lia*
Judge 1	yes	yes	yes	yes
Judge 2	yes	no	yes	no
Judge 3	no	yes	yes	no
Majority	yes	yes	yes	no

Table 1. Discursive dilemma

It is supposed that all the judges accept the rule $lia \leftrightarrow (act \wedge obl)$. Judge 1 accepts both act and obl and, by the connection rule, he accepts lia. Judge 2 accepts act and rejects obl and, by the connection rule, he rejects lia. Finally, judge 3 rejects act and accepts obl and, by the connection rule, he rejects lia. If the three judges apply a majority rule on each proposition then they face a paradox. Indeed, a majority accepts act, a majority accepts obl, a majority accepts the connection rule $lia \leftrightarrow (act \wedge obl)$. But the majority rejects lia. Thus, when majority voting is applied to each single proposition it yields an inconsistent collective set of judgements (see the last row in Table 1). Note that this inconsistency occurs even though the sets of judgements of the individual judges are all consistent.

Let us now show how the discursive dilemma can be formalised in the logic of acceptance.

We first suppose that 1, 2 and 3 *qua* judges of the court accept the connection rule:

$$\neg \mathrm{A}_{123:c}\bot \tag{1}$$

$$\mathrm{A}_{123:c}(lia \leftrightarrow (act \wedge obl)) \tag{2}$$

Then, we have that judge 1 announces that, *qua* judge of the court, he accepts $act \wedge obl$. Judge 2 announces that, *qua* judge of the court, he accepts $act \wedge \neg obl$. Judge 3 announces that, *qua* judge of the court, he accepts $\neg act \wedge obl$. This has the following effect:

$$\mathrm{A}_{123:c}\mathrm{A}_{1:c}(act \wedge obl) \tag{3}$$

$$\mathrm{A}_{123:c}\mathrm{A}_{2:c}(act \wedge \neg obl) \tag{4}$$

$$\mathrm{A}_{123:c}\mathrm{A}_{3:c}(\neg act \wedge obl) \tag{5}$$

Finally, the three judges use a majority principle for each proposition act, obl and lia. This majority principle is formally expressed by the following six hypotheses:

$$\begin{aligned}
\mathbf{Maj} = \quad \{ \; & \mathrm{A}_{123:c} \bigwedge_{i,j \in \{1,2,3\}, i \neq j} ((\mathrm{A}_{i:c} act \wedge \mathrm{A}_{j:c} act) \rightarrow act), \\
& \mathrm{A}_{123:c} \bigwedge_{i,j \in \{1,2,3\}, i \neq j} ((\mathrm{A}_{i:c} \neg act \wedge \mathrm{A}_{j:c} \neg act) \rightarrow \neg act), \\
& \mathrm{A}_{123:c} \bigwedge_{i,j \in \{1,2,3\}, i \neq j} ((\mathrm{A}_{i:c} obl \wedge \mathrm{A}_{j:c} obl) \rightarrow obl), \\
& \mathrm{A}_{123:c} \bigwedge_{i,j \in \{1,2,3\}, i \neq j} ((\mathrm{A}_{i:c} \neg obl \wedge \mathrm{A}_{j:c} \neg obl) \rightarrow \neg obl), \\
& \mathrm{A}_{123:c} \bigwedge_{i,j \in \{1,2,3\}, i \neq j} ((\mathrm{A}_{i:c} lia \wedge \mathrm{A}_{j:c} lia) \rightarrow lia), \\
& \mathrm{A}_{123:c} \bigwedge_{i,j \in \{1,2,3\}, i \neq j} ((\mathrm{A}_{i:c} \neg lia \wedge \mathrm{A}_{j:c} \neg lia) \rightarrow \neg lia) \; \}
\end{aligned}$$

It is possible to prove that 1, 2, 3, 4 and 5 together with **Maj** lead to a contradiction using the axiomatisation of acceptance logic. Indeed, from hypotheses 3, 4 and 5 we infer

$$\mathrm{A}_{123:c}(\mathrm{A}_{1:c}\mathit{act} \wedge \mathrm{A}_{2:c}\mathit{act}) \wedge \mathrm{A}_{123:c}(\mathrm{A}_{1:c}\mathit{obl} \wedge \mathrm{A}_{3:c}\mathit{obl}).$$

By the first and third hypotheses in **Maj**, the latter implies

$$\mathrm{A}_{123:c}\mathit{act} \wedge \mathrm{A}_{123:c}\mathit{obl}$$

and by hypothesis 2 and standard modal principles the latter implies

$$\mathrm{A}_{123:c}\mathit{lia}.$$

From hypotheses 1 and 2, by axioms **4∗** and **5∗**, we can infer

$$\mathrm{A}_{123:c}(\mathrm{A}_{123:c}(\mathit{lia} \leftrightarrow (\mathit{act} \wedge \mathit{obl})) \wedge \neg\mathrm{A}_{123:c}\bot).$$

By Axiom **Inc** and standard modal principles, the latter implies

$$\mathrm{A}_{123:c}(\mathrm{A}_{1:c}(\mathit{lia} \leftrightarrow (\mathit{act} \wedge \mathit{obl}))\wedge$$
$$\mathrm{A}_{2:c}(\mathit{lia} \leftrightarrow (\mathit{act} \wedge \mathit{obl})) \wedge \mathrm{A}_{3:c}(\mathit{lia} \leftrightarrow (\mathit{act} \wedge \mathit{obl}))).$$

From this, by hypotheses 3, 4 and 5 and standard modal principles, we can infer

$$\mathrm{A}_{123:c}(\mathrm{A}_{1:c}\mathit{lia} \wedge \mathrm{A}_{2:c}\neg\mathit{lia} \wedge \mathrm{A}_{3:c}\neg\mathit{lia})$$

and by the sixth hypothesis in **Maj** the latter implies

$$\mathrm{A}_{123:c}\neg\mathit{lia}.$$

Thus, we have $\mathrm{A}_{123:c}\bot$, and by hypothesis 1 we can infer $\bot$.

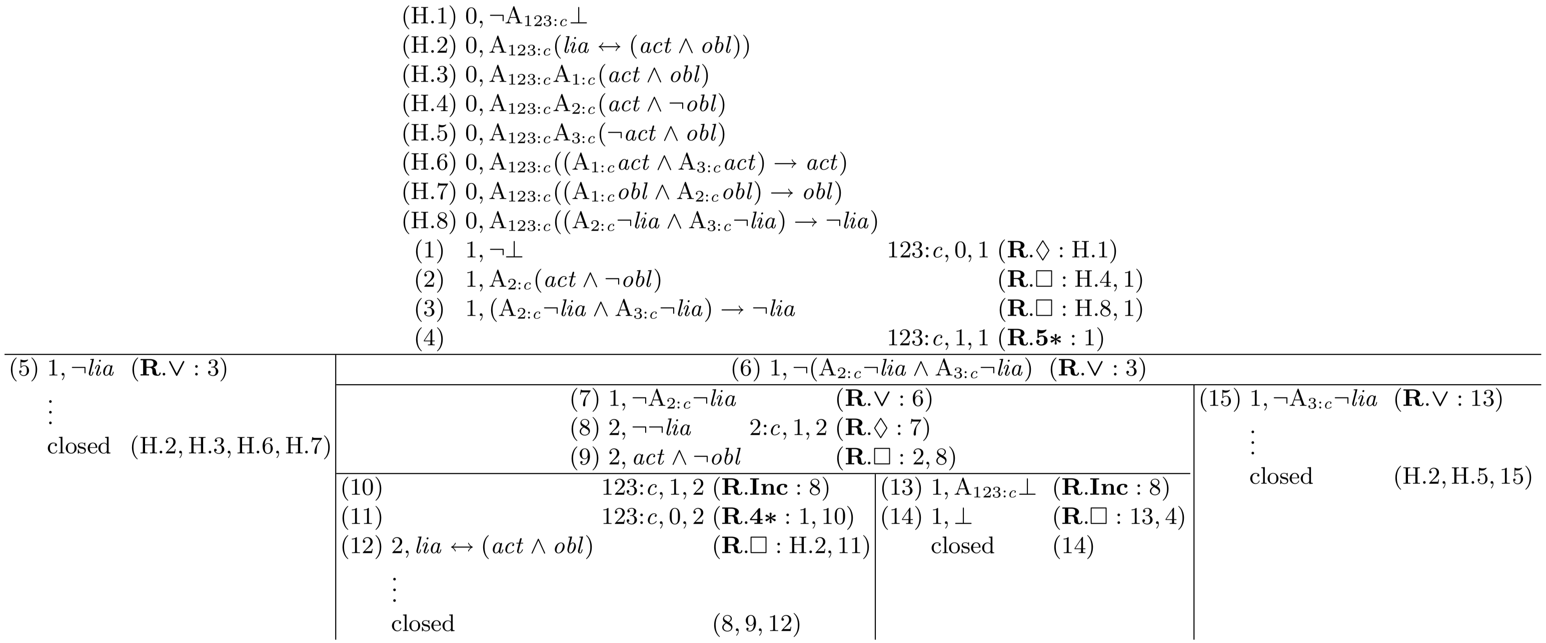
However, if one slightly changes the hypotheses, the proof may change completely. This means that the axiomatisation does not provide a straightforward way to automatise the process of finding such kind of inconsistency. On the other hand, a tableau method is meant to provide a systematic way to search for models and, thereby, an easy way to automatise the process of deciding whether a formula is inconsistent or not.

As an illustration, we show a closed tableau for this example (schematically) in Figure 5. On that figure, the lines identified by H.1–5 correspond to hypotheses 1–5 above, respectively. The line identified by H.6 corresponds to the first hypothesis in **Maj**, H.7 to the third hypothesis, and H.8 to the sixth. The branch containing lines 13 and 14 is already closed. The other branches are not completely displayed. But it is easy to see that all branches generated from line 12 on can be closed, since lines 8, 9 and 12 together are inconsistent in propositional logic. The branches generated from line 15 on can be closed in an analogous way as the latter. And the branches generated from line 5 on need more effort, but they can be closed too, in a similar way as the latter by using lines H.2, H.3, H.6 and H.7.

5 Conclusion

The contribution of this paper is a semantic tableau method for acceptance logic. The method consists in a procedure to check satisfiability of formulae, that can be easily automatised. Given that acceptance logic can be used to formalise some aspects of judgement aggregation, our method also provides an automated reasoning procedure for making judgement aggregation in modal logic.

It is to be noted that, differently from [21, 22], in which logical approaches specialised for judgement aggregation have been proposed, in the logic of accep-



(H.1) $0, \neg \mathrm{A}_{123:c}\bot$

(H.2) $0, \mathrm{A}_{123:c}(lia \leftrightarrow (act \wedge obl))$

(H.3) $0, \mathrm{A}_{123:c}\mathrm{A}_{1:c}(act \wedge obl)$

(H.4) $0, \mathrm{A}_{123:c}\mathrm{A}_{2:c}(act \wedge \neg obl)$

(H.5) $0, \mathrm{A}_{123:c}\mathrm{A}_{3:c}(\neg act \wedge obl)$

(H.6) $0, \mathrm{A}_{123:c}((\mathrm{A}_{1:c} act \wedge \mathrm{A}_{3:c} act) \rightarrow act)$

(H.7) $0, \mathrm{A}_{123:c}((\mathrm{A}_{1:c} obl \wedge \mathrm{A}_{2:c} obl) \rightarrow obl)$

(H.8) $0, \mathrm{A}_{123:c}((\mathrm{A}_{2:c}\neg lia \wedge \mathrm{A}_{3:c}\neg lia) \rightarrow \neg lia)$

(1) $1, \neg\bot$ $123{:}c, 0, 1$ (**R**.$\Diamond$: H.1)

(2) $1, \mathrm{A}_{2:c}(act \wedge \neg obl)$ (**R**.$\Box$: H.4, 1)

(3) $1, (\mathrm{A}_{2:c}\neg lia \wedge \mathrm{A}_{3:c}\neg lia) \rightarrow \neg lia$ (**R**.$\Box$: H.8, 1)

(4) $123{:}c, 1, 1$ (**R**.**5∗** : 1)

(5) $1, \neg lia$ (**R**.$\vee$: 3)

⋮

closed (H.2, H.3, H.6, H.7)

(6) $1, \neg(\mathrm{A}_{2:c}\neg lia \wedge \mathrm{A}_{3:c}\neg lia)$ (**R**.$\vee$: 3)

(7) $1, \neg \mathrm{A}_{2:c}\neg lia$ (**R**.$\vee$: 6)

(8) $2, \neg\neg lia$ $2{:}c, 1, 2$ (**R**.$\Diamond$: 7)

(9) $2, act \wedge \neg obl$ (**R**.$\Box$: 2, 8)

(10) $123{:}c, 1, 2$ (**R**.**Inc** : 8)

(11) $123{:}c, 0, 2$ (**R**.**4∗** : 1, 10)

(12) $2, lia \leftrightarrow (act \wedge obl)$ (**R**.$\Box$: H.2, 11)

⋮

closed (8, 9, 12)

(13) $1, \mathrm{A}_{123:c}\bot$ (**R**.**Inc** : 8)

(14) $1, \bot$ (**R**.$\Box$: 13, 4)

closed (14)

(15) $1, \neg \mathrm{A}_{3:c}\neg lia$ (**R**.$\vee$: 13)

⋮

closed (H.2, H.5, 15)

Fig. 5. A closed tableau for the discursive dilemma

tance judgement aggregation is just an application. We have shown in [2, 3] that the logic of acceptance is a much general formal framework in which the static and dynamic aspects of institutions can be studied (e.g., static and dynamic aspects of social roles, norms and rules).

Moreover, this method uses a rather different set of rules. For instance, rules for axioms **4∗** and **5∗** (which are nothing but a variation of the usual axioms 4 and 5) are similar to the so-called "structural rules" proposed in [23]. Our tableau rules are also modular. This means that it is possible to provide a sound, complete and terminating satisfiability checking method for a logic without one of the semantic constraints, by just removing the corresponding tableau rule. The addition of rules is also possible. For example, one could add the tableau rule:

$$(\mathbf{R}.\mathbf{Mon}) \quad \frac{;\langle G{:}x, n, n'\rangle}{;\langle H{:}x, n, n'\rangle}$$

which corresponds to Axiom **Mon**, thus, obtaining a tableau method for the acceptance logic with Axiom **Mon** proposed in [1, 2].

As possible future works, we intend to investigate computational complexity, and also possible extensions of our method able to address acceptance logic with dynamic operators, such as the logic proposed in [24].

6 Acknowledgements

The contribution by Tiago de Lima is part of the research program Moral Responsibility in R&D Networks, supported by the Netherlands Organisation for Scientific Research (NWO), under grant number 360-20-160.

References

1. Gaudou, B., Longin, D., Lorini, E., Tummolini, L.: Anchoring institutions in agents' attitudes: Towards a logical framework for autonomous MAS. In Padgham, L., Parkes, D.C., eds.: Proceedings of AAMAS'08. (2008) 728–735
2. Lorini, E., Longin, D., Gaudou, B., Herzig, A.: The logic of acceptance: Grounding institutions on agents' attitudes. Journal of Logic and Computation (2009) doi:10.1093/logcom/exn103.
3. Lorini, E., Longin, D.: A logical account of institutions: From acceptances to norms via legislators. In Brewka, G., Lang, J., eds.: Proceedings of KR'08. (2008) 38–48
4. Bratman, M.E.: Practical reasoning and acceptance in context. Mind **101**(401) (1992) 1–15
5. Cohen, L.J.: An essay on belief and acceptance. Oxford University Press, New York, USA (1992)
6. Tuomela, R.: Belief versus Acceptance. Philosophical Explorations **2** (2000) 122–137
7. Tuomela, R.: The Philosophy of Social Practices: A Collective Acceptance View. Cambridge University Press, Cambridge (2002)
8. Boella, G., van der Torre, L.: Norm negotiation in multiagent systems. International Journal of Cooperative Information Systems **16**(1) (2007) 97–122

9. Hart, H.L.A.: The concept of law. Clarendon Press, Oxford (1992) new edition.
10. Gilbert, M.: On Social Facts. Routledge, London and New York (1989)
11. Herzig, A., de Lima, T., Lorini, E.: On the dynamics of institutional agreements. Manuscript (2009)
12. Pettit, P.: Deliberative democracy and the discursive dilemma. Philosophical Issues **11** (2001) 268–99
13. Kornhauser, L.A., Sager, L.G.: Unpacking the court. Yale Law Journal **96** (1986) 82–117
14. Fitting, M.: Proof Methods for Modal and Intuitionistic Logics. Springer (1983)
15. Halpern, J., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence **54** (1992) 311–379
16. Goré, R.: Tableau methods for modal and temporal logics. In D'Agostino, M., Gabbay, D.M., Hahnle, R., Posegga, J., eds.: Handbook of Tableau Methods. Springer (1999) 297–396
17. Pauly, M., van Hees, M.: Logical constraints on judgment aggregation. Journal of Philosophical Logic **35**(6) (2006) 569–585
18. List, C., Pettit, P.: Aggregating sets of judgments: An impossibility result. Economics and Philosophy **18** (2002) 89–110
19. Goldman, A.: Group knowledge versus group rationality: Two approaches to social epistemology. Episteme **1**(1) (2004) 11–22
20. List, C.: Group knowledge and group rationality: A judgment aggregation perspective. Episteme **2**(1) (2005) 25–38
21. Ågotnes, T., van der Hoek, W., Wooldridge, M.: Reasoning about judgment and preference aggregation. In: Proceedings of AAMAS'07. (2007) 566–573
22. Pauly, M.: Axiomatizing collective judgment sets in a minimal logical language. Synthese **158** (2007) 233–250
23. Castilho, M.A., Fariñas del Cerro, L., Gasquet, O., Herzig, A.: Modal tableaux with propagation rules and structural rules. Fundamenta Informaticae **20** (1998) 1–17
24. Herzig, A., de Lima, T., Lorini, E.: What do we accept after an announcement? In Meyer, J.-J.Ch., Broersen, J. eds.: Pre-proceedings of the KR'08-Workshop KRAMAS, pages 81–94, (Available at: http://www.cs.uu.nl/events/kramas2008/PreProceedingsKRAMAS2008.pdf) (2008)