

Mechanical Proof Systems for Logic II, Consensus Programs and Their Processing

*Helena Rasiowa** and *V. Wiktor Marek†*

Abstract

We continue the investigations of [Ra90, Ra91, RM89] and study the automated theorem proving for reasoning about perception of reasoning agents and their consensus reaching. Using the techniques of [Ra91] and of Logic programming ([Ap90, NS93]) we develop the processing techniques for consensus programs.

1 Introduction

Investigations concerning a systematic logical approach to reasoning about knowledge of one or several intelligent agents have in recent years been developed by logicians and computer scientists. These investigations lead to a variety of different logical systems based on various paradigms. We mention here the work of Halpern and Moses ([HM84]), Fagin, Halpern, and Vardi [FHV90]), Mazer ([Ma88]), Orlowska ([Or90]) just to indicate to the reader that the issues of knowledge in the distributive environment are studied widely. TARK proceedings ([Ha86, Va, Pa90, Mo92]) include numerous papers devoted to the subject.

The authors ([RM89]) proposed an approach to reasoning about the perception and the knowledge of groups of fully communicating agents based on the point of view expressed in these principles:

1. The sharpness of agent's perception depends on agent's abilities.
2. Abilities of various agents may be comparable or not comparable (in the sense that one agent may be more capable in one situation, whereas other agent may be more capable in another situation).

*Institute of Mathematics, Warsaw University, Warsaw, Poland. Work partially supported by Polish Government grant KBN 2 2051 91 02. E-mail: harsiowa@mimuw.edu.pl

†Computer Science Department, University of Kentucky, Lexington, KY 40506-0027. Work partially supported by the U.S. National Science Foundation grant IRI-9012902. E-mail: marek@cs.uky.edu

3. Agent's knowledge about a reality is only approximate.
4. Agent's knowledge about a predicate (property) p can be reflected by her perception of p , that is the characteristic features of p acquired in a process of collecting information, conducting research, etc.
5. Agent's knowledge about a property p can be reflected in her abilities to recognize various features and attributes of objects. This may require access to specific recognition medium such as a database, laboratory, test etc.

Starting with this intuition the authors ([RM89]) developed a logic with two types of connectives: perception connectives and knowledge connectives. Perception connectives correspond to the approximation connectives of predicate logic ([Ra87, Ra88, Ra90]). The knowledge connectives correspond to those of the logic of knowledge of Orłowska ([Or90]). The logics without knowledge operators, but with the perception connectives only, have been reexamined by Rasiowa ([Ra91]), under the terms of perception logics. Rasiowa ([Ra91]) established an automated theorem proving technique for such logics. A similar approach has been proposed by Fitting ([Fi92]). In that paper Fitting uses a similar paradigm. The relationship between the experts called *domination* is captured by means of many-valued Kripke models.

In this paper we look again at the perception logics (that is we leave the knowledge part of the language unattended) and prove several results relating the ordinary resolution method of Robinson ([Ro65]) and the form of the resolution discussed in [Ra91].

Our approach here is based on a different technique, much more closely connected to the current presentations of resolution. In this we follow the current texts by Apt ([Ap90]) and Nerode and Shore ([NS93]). The technique used in [Ra91] employed a variant of an argument used by Orłowska in her investigations on the resolution for multivalued logics.

The paper is organized as follows. In the next section we give a short technical introduction to the logic of perception.

In the third section we explore in detail the relationship between the ordinary resolution and T -resolution. We prove a technical result showing the precise connection between these forms of resolution. We also see that a more specialized variant of resolution, the linear input resolution, is preserved by the transformations of the resolution proof trees we use to encode the T -resolution in the ordinary resolution and vice versa.

In the fourth section we outline the theory of logic programming associated with the T -resolution.

In the final section we show how the theory developed in the propositional case can be lifted to the predicate case.

2 Perception logics and resolution

We will now make precise some intuitions described in Section 1.

Let T be a finite set (of reasoning agents). The set T is endowed with a partial ordering \leq_T . Intuitively, $s \leq_T t$ means that the agent t is more perceptive than the agent s . This is interpreted as follows. When both the agents s and t are asked about a specific fact p , the agent t will be less gullible. Her abilities are better in recognizing if the fact p really happens. Specifically, t can find that p did *not*, actually, happen (whereas s perceives p as true). In particular, when $s \leq_T t$ and p is an atomic statement then, whenever t perceives p to be true, s perceives p to be true. In other words, the property $s \leq_T t$ means that this sharper perception happens for all possible facts p . Notice, that a similar property is called “dominance” in Fitting ([Fi92]). All agents observe the same reality. This means that each agent is endowed with a valuation of the set of atoms At (in the propositional case) or a relational structure (with the same underlying algebra, that is with the same objects, and the same interpretation of the function symbols and the constants) – in the predicate case.

Denoting by V_t the valuation assigned to the agent t the requirement of better perception for a stronger agent is formally expressed as

$$w \leq_T t \Rightarrow \forall_{p \in At} V_t(p) \leq V_w(p) \quad (1)$$

In the predicate case, denoting by \mathcal{A}_t the relational system assigned as the perception of the agent t we have, for every predicate letter p ,

$$w \leq_t t \Rightarrow p^{\mathcal{A}_t} \subseteq p^{\mathcal{A}_w} \quad (2)$$

Formally, the language of perception logic is the language of the classical logic extended by unary modal operators d_t , for $t \in T$. Intuitively, $d_t\varphi$ means that the agent t (and, as will turn out all the agents with weaker perception) perceives φ .

The semantics for our language is determined by the consensus interpretation. In the predicate case a T -reality for the underlying language is a collection of first order relational structures. The collection is indexed by the set T . All the structures in a T -reality have the same underlying algebra and the T -reality must satisfy Condition (2).

Analogously we define the notion of T -reality for a propositional case. It is a collection of valuations of the underlying set of atoms. Moreover we require (1).

Once it is clear what we mean by the realities for our language, we define the notion of satisfaction. The fact that a T -reality $\mathcal{M} = \langle \mathcal{A}_t \rangle_{t \in T}$ satisfies φ means that the consensus about φ has been reached. This, of course implies that the set of formulas true in a T -reality is *not* complete. The notion of satisfaction for formulas is defined in a roundabout way. We give the full definition of satisfaction for the predicate case. The propositional

case can be easily described by an obvious modification of the clause (a) and eliminating quantifier cases (g) and (h). First we define the relation $\mathcal{M} \models d_t(\varphi)[v]$ (where v is a valuation of variables).

- (a) $\mathcal{M} \models d_t(p_i(x_1, \dots, x_m))[v]$ iff $\mathcal{A}_t \models p_i(x_1, \dots, x_m)[v]$
- (b) $\mathcal{M} \models d_t(\phi \vee \psi)[v]$ iff $\mathcal{M} \models d_t(\phi)[v]$ or $\mathcal{M} \models d_t(\psi)[v]$
- (c) $\mathcal{M} \models d_t(\phi \wedge \psi)[v]$ iff $\mathcal{M} \models d_t(\phi)[v]$ and $\mathcal{M} \models d_t(\psi)[v]$
- (d) $\mathcal{M} \models d_t(\phi \Rightarrow \psi)[v]$ iff for all $s \leq_T t$, $\mathcal{M} \models d_s(\phi)[v]$ implies $\mathcal{M} \models d_s(\psi)[v]$
- (e) $\mathcal{M} \models d_t(\neg\phi)[v]$ iff for all $s \leq_T t$, not $(\mathcal{M} \models d_s(\phi)[v])$
- (f) $\mathcal{M} \models d_s(d_t(\phi))[v]$ iff $\mathcal{M} \models d_t(\phi)[v]$
- (g) $\mathcal{M} \models d_t(\forall x_i \phi)[v]$ iff for all $a \in M$, $\mathcal{M} \models d_t(\phi)[v(i/a)]$
- (h) $\mathcal{M} \models d_t(\exists x_i \phi)[v]$ iff there exists $a \in M$, $\mathcal{M} \models d_t(\phi)[v(i/a)]$

Next, we define the satisfaction for all formulas of the underlying language L .

$$\mathcal{M} \models \Psi[v] \text{ if and only if } \mathcal{M} \models d_t(\Psi)[v] \text{ for all } t \in T.$$

A complete axiomatization for the relation \models has been given in [RM89].

Here is a short list of fundamental properties of the satisfaction (consensus) relation for a T -reality \mathcal{M} .

$$\mathcal{M} \models \neg d_t(\varphi)[v] \text{ iff } \text{not } \mathcal{M} \models d_t(\varphi)[v] \tag{3}$$

$$\mathcal{M} \models d_w(\neg d_t(\varphi))[v] \text{ iff } \mathcal{M} \models \neg d_t(\varphi)[v] \tag{4}$$

$$\mathcal{M} \models (d_t(\varphi) \Rightarrow d_t(\psi))[v] \text{ iff } \text{not } \mathcal{M} \models d_t(\varphi)[v] \text{ or } \mathcal{M} \models d_t(\psi)[v] \tag{5}$$

$$\forall_{w, t \in T} w \leq_T t, \mathcal{M} \models d_t(\varphi)[v] \text{ implies } \mathcal{M} \models d_w(\varphi)[v] \tag{6}$$

$$\forall_{w, t \in T} w \leq_T t \text{ implies } \mathcal{M} \models (d_t(\varphi) \Rightarrow d_w(\varphi))[v] \tag{7}$$

Let us shortly describe what these properties mean. The properties (3), (4) and the point (f) of the definition of the satisfaction relation tell us that the agents possess the complete information about the other agents perception. This does not change their perception, though. In particular, the agent's perception of basic facts does not depend on other agents perception. The property (5) tells us that the implication between the statements about perception behaves "classically" (although, in general, implication behaves "intuitionistically" in T -realities). The properties (6) and (7) tell us that the perception behaves antimonotonically with respect to \leq_T .

These properties and the fact that the quantifiers commute with the operators d_t allow us to find the prenex form for formulas of the language. Next, we can perform Skolemization of the prenex form of the formulas $d_t\varphi$. Finally, using the properties (b), (c) and (f) in the definition of satisfaction, and the property (4), we can eliminate nested occurrences of d_t .

Let us introduce the notion of *D-atom*. It is a formula of the form $d_t p(s_1, \dots, s_r)$ where $t \in T$ and s_1, \dots, s_r are terms of the language (in the propositional case no term is present). A *D-literal* is an *D-atom* or its negation. A *D-clause* is a disjunction of *D-literals*.

Summing up all properties of formulas of the form $d_t\varphi$, we quote the following result of [Ra91]

Proposition 1 *For every formula of the form $d_t\varphi$ there is a finite set of D-clauses S such that $d_t\varphi$ and S are equisatisfiable. That is there is a T -reality \mathcal{M} satisfying $d_t\varphi$ if and only if there is a T -reality \mathcal{M}' such that \mathcal{M}' satisfies all the D -clauses from S .*

Proposition suggests that we may be facing a situation similar to that encountered in automated theorem proving. A version of resolution principle may work here. Indeed it is the case. In [Ra91] Rasiowa found a variant of the resolution principle suitable for our context. We will refer to this rule as *T-resolution rule*. We shall denote this rule by res_T .

$$\frac{A' \cup \{d_t(p(t_1, \dots, t_k))\}, \quad A'' \cup \{\neg d_w(p(s_1, \dots, s_k))\}}{(A' \cup A'')\Theta} \quad \text{providing } w \leq_T t \quad (8)$$

Notice the asymmetry of the rule res_T . Its applicability is restricted by the condition $w \leq_T t$. Here we assume that the parent clauses are standardized apart. Θ is a most general unifier of atoms $p(t_1, \dots, t_k)$ and $p(s_1, \dots, s_k)$.

As usual, by Herbrand *T*-reality we mean a *T*-reality whose underlying universe consists of ground terms of the language.

The following result is proved in [Ra91]

Proposition 2 *Let S be a set of D-clauses. Then there is a T -reality satisfying S if and only if the closure of S under T -resolution does not contain an empty clause.*

In the next section we shall examine the principle of *T*-resolution and show how it relates to ordinary resolution.

3 Automated theorem proving for consensus reaching

In this section we reexamine the automated theorem proving method for consensus reaching. The automated theorem proving system for consensus reaching is based on the version

of resolution rule valid for our system. Here we show how the rule res_T can be used for actual processing of consensus logic programs.

We will prove a basic result on the relationship of the asymmetric resolution rule introduced in [Ra91] and the usual resolution rule (for best description see Nerode and Shore [NS93]).

We will study now the propositional case. In section 5 we will lift these results to the predicate case. Let $\langle T, \leq_T \rangle$ be a poset, and let At be a set of atoms. Recall that a D -atom is an expression of the form $d_t(p)$ where $p \in At$, $t \in T$. Similarly, a D -literal is a D -atom or its negation. Next, a D -clause is a finite set of D -literals. As usual, \square is the empty clause (interpreted as falsity).

In our context, the T -resolution rule is the following rule of proof:

$$res_T \quad \frac{A' \cup \{d_t(p)\}, \quad A'' \cup \{\neg d_w(p)\}}{A' \cup A''}$$

where $w \leq_T t$.

The ordinary resolution rule in our setting takes this form:

$$res \quad \frac{A' \cup \{d_t(p)\}, \quad A'' \cup \{\neg d_t(p)\}}{A' \cup A''}$$

The T -resolution rule is asymmetric. We can resolve on a D -atom $d_t(p)$ against $\neg d_w(p)$ only if $w \leq_T t$. Hence, D -clauses $\{d_w(p)\}$ and $\{\neg d_t(p)\}$ ($w \leq_T t$) do not entail \square . This agrees with our paradigm; since $w \leq_T t$, the atom p may be perceived by the agent w as true and the agent t as false without the contradiction.

Let S be a set of D -clauses. By $\mathcal{R}_T(S)$ we mean the closure of the set S under the rule res_T . Similarly, $\mathcal{R}(S)$ is the closure of S under the usual resolution rule. Notice that res_T is a generalization of the ordinary resolution rule. Indeed, every derivation using the ordinary resolution is a valid res_T reasoning. This is because the relation \leq_T is reflexive. As noticed above, the converse does not necessarily hold. That is, a refutation using T -resolution does not need to be a resolution refutation.

Denote by D_T , the *diagram of T*, the following set of D -clauses:

$$D_T = \{\{\neg d_t(p) \vee d_w(p)\} : \quad w \leq_T t, p \in At\}$$

A more intuitive representation for D_T is:

$$D_T = \{d_t(p) \Rightarrow d_w(p) : \quad w \leq_T t, p \in At\}$$

The set D_T codifies our knowledge about the relationship between the agents. If the agent t is more perceptive than the agent w (which in our system is encoded by $w \leq_T t$), and t accepts a fact p then, certainly, w accepts the fact p - but not vice versa.

We have now the basic result on the connection between the ordinary resolution rule and the T -resolution rule. This result is fundamental for the rest of the paper. Once we prove it, we will be able to list the most of the results of ordinary automated theorem proving with resolution to the case of D -clauses and T -resolution.

Theorem 1 *Let S be a set of D -clauses. Then $\square \in \mathcal{R}_T(S)$ if and only if $\square \in \mathcal{R}(S \cup D_T)$.*

Proof: (\Rightarrow). Assume that τ is a \mathcal{R}_T -refutation of S . We shall transform the tree τ to a tree τ' such that the leaves of τ' are either in S or in D_T , and such that:

1. τ' is a refutation of $S \cup D_T$, and
2. τ' applies only the ordinary resolution rule res .

To this end, let

$$\frac{A' \cup \{d_t(p)\}, \quad A'' \cup \{\neg d_w(p)\}}{A' \cup A''}$$

be an application of T -resolution rule within the tree τ . We transform this occurrence of res_T to the following two applications of the ordinary resolution rule (with an additional input from D_T)

$$\frac{\frac{A' \cup \{d_t(p)\} \quad \{\neg d_t(p), d_w(p)\}}{A' \cup \{d_w(p)\}} \quad A'' \cup \{\neg d_w(p)\}}{A' \cup A''}$$

It should be clear that our transformation $\tau \mapsto \tau'$ is defined by induction on the complexity of the tree τ . At each application of the rule res_T we add one additional leaf from D_T and transform the application of the rule res_T into two applications of the ordinary resolution rule. This implies that the size of the tree τ' is linear in the size of τ . Since no new application of the rule res_T is introduced, it is clear that we can eliminate in the manner outlined above, all the applications of res_T . The resulting tree τ' uses only the ordinary resolution rule res . Now, notice that all the nodes (both the leaves and the inner nodes) of the tree τ appear in the tree τ' . Therefore \square appears in τ' and so τ' is a resolution refutation. Finally, notice that the only new leaves we introduced come from D_T . Therefore τ' is a refutation of $S \cup D_T$. Thus (\Rightarrow) is proved.

(\Leftarrow). Conversely, we show that if there is a resolution refutation of $S \cup D_T$, then there is a T -resolution refutation of S alone. The construction will be more involved in this case. Our strategy is to transform a resolution refutation of $S \cup D_T$ to a T -resolution refutation of S alone by a consecutive elimination of D -clauses from D_T (and introducing more applications of the rule res_T). The argument will be again inductive, but the induction will be over a different argument. We will be consecutively eliminating the *deepest* application of the resolution whose one of the parents is a D -clause from D_T . By the deepest application we mean here the application the least distanced from the root.

Notice that since $\langle T, \leq_T \rangle$ is a poset, therefore, if τ is a resolution refutation out of S alone, then τ is a T -resolution refutation from S .

Now, assume that τ has a certain number of leaves from D_T . We shall show how to eliminate the last application of the resolution with a parent in D_T . This will be done at a cost of using T -resolution at a different place, and the reconstruction of the refutation tree.

Hence, let $C = \{\neg d_t(p), d_w(p)\}$, $w \leq_T t$ be the D -clause used in τ at some leaf l . Moreover assume that the distance of l from the root is least among such leaves.

Two situations are possible. In both the D -clause C is an input to the (ordinary) resolution. However we resolve either on $d_t(p)$ or on $d_w(p)$.

Case 1. We resolve on $d_t(p)$. Then the application of the ordinary resolution is this:

$$\frac{A \cup \{d_t(p)\} \quad \{\neg d_t(p), d_w(p)\}}{A \cup \{d_w(p)\}}$$

Since τ is a refutation, τ must contain a descendant of $D = A \cup \{d_w(p)\}$, say $D' = A' \cup \{d_w(p)\}$, and a D -clause $E = A'' \cup \{\neg d_w(p)\}$, and an application of the (ordinary) resolution

$$\frac{A' \cup \{d_w(p)\} \quad A'' \cup \{\neg d_w(p)\}}{A' \cup A''}$$

Thus this application of the resolution resolves D' against E . We can assume that all the applications of the resolution rule between C and $A' \cup A''$ resolve on D -literals different from $d_w(p)$. Now we modify our tree τ as follows. We *do not* resolve $A \cup \{d_t(p)\}$ against C . We resolve $A \cup \{d_t(p)\}$ against the further D -clauses in the tree τ and get, instead of $A' \cup \{d_w(p)\}$, $A' \cup \{d_t(p)\}$. Now we resolve the D -clause $A' \cup \{d_t(p)\}$ against $A'' \cup \{\neg d_w(p)\}$ using the T -resolution. Again $A' \cup A''$ is obtained. The resulting tree τ' has one leaf from D_T less and one application of T -resolution more.

Case 2. We resolve on $\neg d_w(p)$. Then the application of the ordinary resolution is this:

$$\frac{A \cup \{\neg d_w(p)\} \quad \{\neg d_t(p), d_w(p)\}}{A \cup \{\neg d_t(p)\}}$$

Since τ is a refutation, there must be a descendant of $D = A \cup \{\neg d_t(p)\}$, say $D' = A' \cup \{\neg d_t(p)\}$, and a D -clause $A'' \cup \{d_t(p)\}$, and an application of the (ordinary) resolution

$$\frac{A' \cup \{\neg d_t(p)\} \quad A'' \cup \{d_t(p)\}}{A' \cup A''}$$

We can assume that all the applications of the resolution rule between C and $A' \cup A''$ resolve on literals different from $d_t(p)$. Now we modify our tree τ as follows. We *do not* resolve $A \cup \{\neg d_w(p)\}$ against C . We resolve $A \cup \{\neg d_w(p)\}$ against the further D -clauses in the tree τ and get, instead of $A' \cup \{\neg d_t(p)\}$, $A' \cup \{\neg d_w(p)\}$. Now we resolve the D -clause

$A' \cup \{\neg d_w(p)\}$ against $A'' \cup \{d_t(p)\}$ using the T -resolution. Again $A' \cup A''$ is obtained. The resulting tree τ' has one leaf from D_T less and one application of T -resolution more.

Thus, by iteration of the construction described above we can eliminate the use of all D -clauses from D_T appearing in the leaves of τ . The resulting tree τ^* shares with τ the property that all the D -literals eliminated by the resolution in τ are also eliminated in τ^* . Since all the D -literals were eliminated in τ (after all τ is a refutation), τ^* is also a refutation. \square

Figures 1 and 2. illustrate the transformation of the second part of our theorem.

Proposition 3 *Let S be a set of D -clauses. Then there is a T -reality satisfying S if and only if $\square \notin \mathcal{R}(S \cup D_T)$.*

Proof: If \mathcal{M} is a T -reality satisfying S then, since D_T is satisfied in every T -reality, \mathcal{M} satisfies $S \cup D_T$. Therefore \mathcal{M} satisfies $\mathcal{R}(S \cup D_T)$. Thus $\square \notin \mathcal{R}(S \cup D_T)$.

Conversely, if $\mathcal{R}(S \cup D_T)$ does not contain \square then there is a model of $S \cup D_T$. Since this model satisfies D_T , we can read off this model a T -reality satisfying S . \square

Corollary 1 ([Ra91]) *Let S be a set of D -clauses. Then there is a T -reality satisfying S if and only if $\square \notin \mathcal{R}_T(S)$.*

Let us restrict now to the case of Horn D -clauses. Here, the important observation is that all the D -clauses in the diagram of T , D_T , are Horn D -clauses. Recall that a Horn D -clause is a D -clause that contains at most one positive literal. In our setting Horn D -clauses are of the form:

$$\{d_s(p), \neg d_{w_1}(q_1), \dots, \neg d_{w_k}(q_k)\}$$

or of the form:

$$\{\neg d_{w_1}(q_1), \dots, \neg d_{w_k}(q_k)\}$$

The D -clause $\{d_s(p), \neg d_{w_1}(q_1), \dots, \neg d_{w_k}(q_k)\}$ is called a T -program clause and usually denoted by

$$d_s(p) \leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k)$$

The second type of Horn D -clause is called a *goal* and is denoted in the logic programming by:

$$\leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k)$$

For the notion of linear input resolution see Nerode and Shore [NS93]. We can consider the linear input resolution in our context. That is, we consider a linear tree starting with a goal and applying T -resolution instead of ordinary resolution.

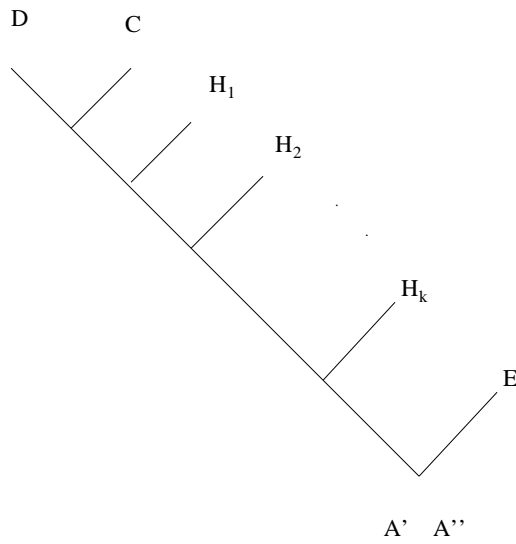


Figure 1: A fragment of the tree τ before the transformation of the second part of the theorem

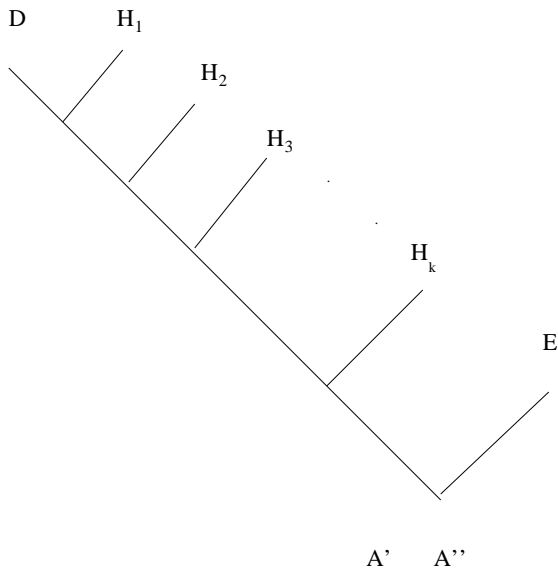


Figure 2: A fragment of the tree τ' after the transformation

The crucial observation now is that not only D_T consists of Horn clauses, but the transformations described in both parts of our Theorem 1 preserve the linear input resolution.

Specifically we have:

Theorem 2 *Let S be a set of Horn D -clauses. Then there is no T -reality satisfying S if and only if S possesses a linear input refutation using the T -resolution rule.*

Proof: We look carefully at the transformations described in the proof of Theorem 1.

Assume that no T -reality satisfies S . Then by Corollary 1 S has a T -resolution refutation. Therefore $S \cup D_T$ possesses an (ordinary) resolution refutation. Hence $S \cup D_T$ possesses an (ordinary) linear input refutation. Now we use the transformation of the second part of Theorem 1. The resulting tree is again a linear input refutation, but uses the T -resolution. Thus S possesses a linear input refutation using T -resolution rule.

Conversely, since a linear input refutation is a refutation, therefore if S possesses a linear input refutation using T -resolution rule then S possesses a T -refutation. Therefore no T -reality satisfies S . \square

Corollary 2 *Let S be a theory consisting of Horn D -clauses. The following are equivalent:*

1. S possesses a refutation using T -resolution rule.
2. S possesses a linear input refutation using T -resolution rule.
3. $S \cup D_T$ possesses a linear input refutation using ordinary resolution rule.
4. $S \cup D_T$ possesses a refutation using ordinary resolution rule.
5. $S \cup D_T$ is unsatisfiable.
6. There is no T -reality satisfying S .

4 Consensus Programs and their processing

In this section we discuss T -programs and their processing. Recall from Section 3 that a T -program clause is a D -clause of the form

$$C = d_s(p) \leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k)$$

A T -program is any set of T -program clauses. Intuitively, a T -program P describes a T -reality, with various interconnections of agent perceptions. Intuitively, the T program clause C tells us that in the T -reality described by P this happens: whenever the agent w_1 perceives the fact q_1 , and the agent w_2 perceives the fact q_2 etc. then the agent s perceives p .

4.1 The least T -reality for a program

The valuations of D -atoms can be put into a one-to-one correspondence with the subsets of D -atoms in the usual fashion. Therefore we will not distinguish between the valuations of D -atoms and sets of D -atoms. Also, it is obvious that T -realities are naturally ordered by inclusion.

Proposition 4 *For every T -program P there exists the least T -reality satisfying P .*

Proof: As in the classical case it is easy to see that the intersection of all T -realities satisfying P is also a T -reality satisfying P . Therefore this intersection is the least T -reality satisfying P . \square

We will describe now a sound and complete method of processing queries to T -programs.

The way we are going to process our queries is a variant of the usual processing of logic program and will reflect precisely the difference between the usual resolution rule and the T -resolution rule. To set a terminology, we shall call this operation T -matching. Formally, a D -atom $d_s(p)$ T -matches a T program clause $C = d_t(p) \leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k)$ if $s \leq_T t$. It should be clear that when $s = t$ then matching reduces to selection of an D -atom $d_t(p)$ for expansion.

The second part of the processing procedure coincides with that of ordinary logic programming. Once we have a goal $\leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k)$ and select within it a D -atom $d_{w_i}(q_i)$ and the D -atom $d_{w_i}(q_i)$ T -matches a T program clause

$$d_t(q_i) \leftarrow d_{z_1}(h_1), \dots, d_{z_n}(h_n)$$

then we create a new goal

$$\leftarrow d_{w_1}(q_1), \dots, d_{w_{i-1}}(q_{i-1}), d_{z_1}(h_1), \dots, d_{z_n}(h_n), d_{w_{i+1}}(q_{i+1}), \dots, d_{w_k}(q_k)$$

this process is called *expansion*. Hence, the expansion process corresponds to one application of T -resolution between the current goal and some T program clause in P .

Let G be a goal. We say that the goal G succeeds if there is a sequence of goals G_0, \dots, G_m such that $G_0 = G$, $G_m = \square$ and each G_{i+1} arises from G_i by selecting a D -atom in G_i , matching it with some T program clause in P and expanding.

We now have the following result.

Proposition 5 *Let P be a T -program. Let G be a goal. Then there is no T -reality satisfying $P \cup \{G\}$ if and only if the goal G succeeds.*

Proof: As mentioned above the procedure of T -matching and expanding corresponds to the linear input T -resolution. Therefore our result follows from Theorem 2. \square

Let us look at an example.

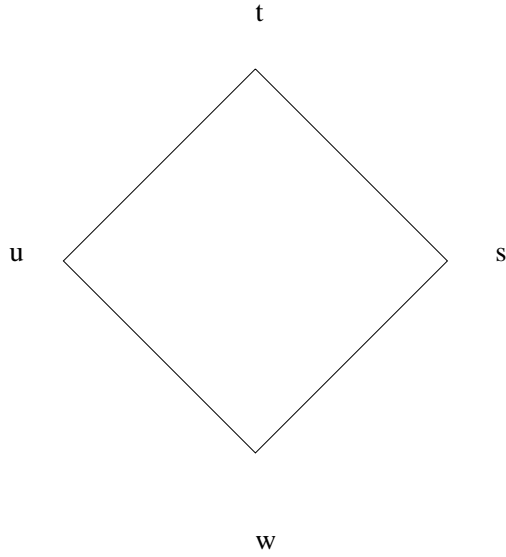


Figure 3: Partial Ordering T

Example 1 Let T be the partial ordering of Figure 3. Consider the following simple T -program

$$\begin{aligned} d_u(p) &\leftarrow d_w(q), d_s(r) \\ d_t(r) &\leftarrow \\ d_s(q) &\leftarrow \end{aligned}$$

Given the goal $\leftarrow d_w(p)$ we notice that the D -atom in that goal T -matches the first clause in our program. It is not identical with it, just T -matches it, because $w \leq_T u$. This creates by expansion a new goal $\leftarrow d_w(q), d_s(r)$. The first D -atom in this goal matches the second clause in our program because $w \leq t$. The expansion creates now the goal $\leftarrow d_s(r)$. The D -atom in this goal matches the third clause. The result of expansion is now \square and so the original goal succeeds.

The completeness result (Proposition 5) implies the following proposition.

Proposition 6 Let $d_s(p)$ be a D -atom. Let P be a T -program. Then the goal $\leftarrow d_s(p)$ succeeds if and only if $d_s(p)$ belongs to the least T -reality satisfying P .

Proof: The goal $\leftarrow d_s(p)$ succeeds if and only if $P \cup \{\neg d_s(p)\}$ is not satisfiable. This is equivalent precisely when $d_s(p)$ belongs to the least T -reality satisfying P . \square

The operator T_P associated to logic program can be lifted to the present situation with some modifications. The difference is that as we compute new D -atoms, we also need to add D -atoms perceived by less perceptive agents.

Specifically, given a T -program P define an operator S_P as the T -closure of the set

$$\{d_w(p) : \text{There exists } C \in P, C = d_s(p) \leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k), \text{ and } \\ d_{w_1}(q_1) \in M, \dots, d_{w_k}(q_k) \in M, w \leq_T s\}.$$

Proposition 7 1. *The operator S_P is monotone and finitizable.*

2. *Hence, S_P possesses the least fixpoint which is equal to $\bigcup_{n \in \omega} S_P^n(\emptyset)$.*

Similarly to the case of ordinary logic programs we have the following theorem.

Theorem 3 *Let P be a T -program. Then*

1. *The least T -reality satisfying P coincides with the least fixpoint of S_P .*
2. *The least fixpoint of S_P coincides with the set of D -atoms $d_s(p)$ for which the goal $\leftarrow d_s(p)$ succeeds.*

Proof: (1) Since T -realities of theories are always T -closed it is only necessary to see that for a T -reality satisfying P the set $\{d_w(p) : \text{There exists } C \in P, d_s(p) \leftarrow d_{w_1}(q_1), \dots, d_{w_k}(q_k), \text{ and } d_{w_1}(q_1) \in M, \dots, d_{w_k}(q_k) \in M, w \leq_T s\}$ is included in M . This is routinely checked. This implies that the least fixpoint of S_P is included in every T -reality satisfying P , hence it is included in the least T -reality satisfying P . On the other hand it is easy to see that the least fixpoint of S_P is itself a T -reality satisfying P . This completes the proof of (1). (2) follows from Proposition 6 and (1). \square

4.2 Processing the consensus queries

Now, it is clear how we can get a result about processing atomic queries *not* involving the operator d_t for T -programs. Such query to a program P is a query about *consensus* in the T -reality described by the program P . To get the answer to such query, say p (where p) is an atom, we must check if *all* D -atoms $d_t(p)$ succeed. It turns out that we do not need to check all atoms $d_t(p)$. It follows immediately from the basic property (7) of T -realities (see Section 2) that it is enough to check if the queries $d_t(p)$ succeed for all the *maximal* elements t of T . Similarly, if a query $d_w(p)$ fails for every *minimal* element s of T then the T -reality described by our program P satisfies $\neg p$.

At a bigger cost we can now process an arbitrary consensus query. Given a propositional formula φ , the formula φ is satisfied in the least T -reality \mathcal{M}_T satisfying the program P if and only if all formulas $d_t(\varphi)$ are satisfied in \mathcal{M}_T . Assume for a moment that φ does not contain the implication functor, and that the negation functor appears only in front of expression of the form $d_s(\psi)$. Clearly, the formula $d_t(\varphi)$ is logically equivalent to a set of

D -clauses. Thus we need to be able to check whether a D -clause is satisfied in \mathcal{M}_T . But such D -clause C is satisfied in \mathcal{M}_T if and only if one of D -literals in C is satisfied in \mathcal{M}_T . This, together with the above remarks on testing the validity of D -literals in \mathcal{M}_T , gives a method for testing consensus for *arbitrary* propositional formulas.

Finally, if φ does contain implication and unrestricted negation, then, again, we can test the consensus about φ , but now the cost is bigger. This happens because in the recursive definition of satisfaction we need to consult the perception of less perceptive agents.

5 Predicate T -resolution and T -programs

Let us now look at the predicate case. The T -resolution is similar to the ordinary resolution in that the propositional Herbrand refutations can be lifted to the full predicate case. Specifically, we have

Proposition 8 *Let S be a set of predicate D -clauses. Let S^+ be the set of ground substitutions of D -clauses from S . Then, if τ^+ is a T -resolution proof of a D -clause C^+ from S^+ then there is a D -clause C and a T -resolution proof τ of the D -clause C from S and there exists a substitution Θ such that $\tau\Theta = \tau^+$ and $C\Theta = C^+$.*

Proof: The usual proof by induction on the depth of resolution tree is easily adopted to our case. \square

Corollary 3 ([Ra91]) *Let S be a set of D -clauses. Then there is no T -reality satisfying S if and only if $\square \in \mathcal{R}_T(S)$.*

Proof: Clearly, S is T -satisfiable if and only if there is a Herbrand T -reality satisfying T (the usual argument can be used in this case). This implies that $\square \notin \mathcal{R}_T$.

Conversely, if $\square \notin \mathcal{R}_T(S)$ then by lifting (Proposition 8), $\square \notin \mathcal{R}_T(S^+)$. Then, by Corollary 1 there is a propositional T -reality satisfying S^+ . We can read off from such T -reality a (predicate) Herbrand T -reality satisfying S . \square

Next, we look at the predicate T -programs. Since the “lifting” operation does not change the form of the resolution tree, the linear input T -resolution with unification is sound and complete for processing the goals in T -programs.

As in the case of ordinary predicate resolution the T -predicate resolution can be applied only if the D -atom $d_w(p)$ selected for expansion *unify* with a head $d_t(q)$ of a T program clause in P . The unification requires now that two facts happen:

1. p unifies with q

2. $w \leq_T t$

The first requirement can be processed by means of the usual unification algorithm (see [NS93]). The second requires maintaining a database of the diagram of T and checking against it for applicability.

Large body of Logic Programming lifts to the context of T -resolution and T -programs. We will not develop such theory in this paper, but will quote just one result, indicative to the flavor of many facts that can be proven.

Proposition 9 *Let $d_t p(s_1, \dots, s_k)$ be a ground D -atom of L . Let P be a T -program. Then the query $\leftarrow d_t p(s_1, \dots, s_k)$ succeeds if and only if the least Herbrand T -reality for P satisfies $d_t p(s_1, \dots, s_k)$.*

6 Conclusions and further research

So far we were able to study the automated theorem proving for the perception part of the logic of perception and knowledge introduced in [RM89]. Extending these results to the general case (the presence of knowledge modality) requires overcoming of serious technical obstacles. We hope that this will become possible during further investigations. Especially important would be incorporation of knowledge modality to the Horn fragment of the perception logic without forfeiting nice processing capabilities of that fragment.

References

- [Ap90] K. Apt (1990), “Logic Programming”, In: *Handbook of Theoretical Computer Science*, J. van Leeuwen ed., pp. 493–574, MIT Press, Cambridge, MA.
- [FHV90] Fagin, R., Halpern, J. Y., Vardi, M. (1988), “Model-theoretical Analysis of Knowledge”, IBM Research report RJ 6461.
- [Fi92] Fitting, M. (1992), Many-valued Modal Logics II, *Fundamenta Informaticae* 17, pp. 55-73.
- [Ha86] Halpern, J. Y. (1986) (ed), *Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann.
- [HM84] Halpern, J.Y., Moses, Y. (1984), “Toward a Theory of Knowledge and Ignorance: Preliminary Report”, *Proceedings of AAAI Workshop on Non-Monotonic Reasoning*, pp. 125-143.

- [Ma88] Mazer, M.S. (1988), “A Knowledge Theoretic Account of Recovery in Distributed Systems”, *TARK '88*, M. Vardi ed., pp. 309-324.
- [Mo92] Moses, Y., (1992) (ed), *Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann.
- [NS93] Nerode, A., Shore, R. (1993), *Logic for Applications*, Springer-Verlag.
- [Or85] Orłowska, E. (1985), “Mechanical proof methods for Post logics”, *Logique Anal. N.S.*, 28 Annee, 110-111, 173-192.
- [Or90] Orłowska, E. (1990), “Logic for Reasoning about Knowledge”, *Z. Math. Logik Grund. Math.*
- [OS86] Orłowska, E., Sanders, J. (1986), “Knowledge Transfer in Distributed Systems”, unpublished manuscript.
- [Pa90] Parikh, R. (1990) (ed), *Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann, 1990.
- [Pa82] Pawlak, Z. (1982), “Rough Sets”, *International Journal of Computer And Information Sciences* 11, pp. 341-356.
- [Ra86] Rasiowa, H. (1986), “Rough Concepts and ω^+ -valued Logic”, *Proceedings ISMVL '86*. IEEE Press, pp. 282-289.
- [Ra87] Rasiowa, H. (1987), “Algebraic Approach to Some Approximate Reasonings”, *Proceedings ISMVL '87*. IEEE Press, pp. 342-347.
- [Ra88] Rasiowa, H. (1988), “Logic of Approximation Reasoning”, *Proceedings of CSL '87*. Springer LN in Computer Science 329, pp. 188-210.
- [Ra90] Rasiowa, H. (1990), “On approximation logics: A survey”, Kurt Gödel Gesellschaft, Jahrbuch 1990, pp. 63-87.
- [Ra91] Rasiowa, H. (1991), “Mechanical Proof Systems for Logic: Reaching Consensus by Groups of Intelligent Agents”, *International Journal of Approximate Reasoning* 5, pp. 415-432.
- [RM89] Rasiowa, H., Marek, W. (1989), “On reaching consensus by groups of intelligent agents”. *Proceedings ISMIS'89*, pp. 234-243, North-Holland.
- [RS63] Rasiowa, H., Sikorski, R. (1963), “The Mathematics of Metamathematics”, PWN, Warszawa, (3rd ed. 1970).
- [Ro65] Robinson, G.A. (1965), “A machine oriented logic based on the resolution principle”, *Journal of the Association for Computing Machinery* 12, pp. 23-41.

- [Va] Vardi, M. Y. (1988), (ed), *Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann.

Authors' profiles

- Helena Rasiowa received her M.A. and Ph.D. degrees from the University of Warsaw, Poland. She is a Professor of Mathematics at the University of Warsaw and serves as the Head of Mathematical Logic Section since 1970. She was the Dean of the Faculty of Mathematics, Mechanics and Informatics (1958-60, 1962-66, 1968-76) and served as the President of the Scientific Council of the Institute of Computer Science of the Polish Academy of Sciences (1972-1983). She is an author of about 100 papers in the area of mathematical logic and theoretical computer science including books: *The Mathematics and Metamathematics* (with Roman Sikorski), *Introduction to Modern Mathematics* (North Holland), *An Algebraic Approach to Non-Classical Logics* (North Holland). She is the Editor-in-Chief of *Fundamenta Informaticae Journal*. She also received a Distinguished Teacher Award in Poland (1982) and the First Class State Science Prize in Mathematics (1984). She was the President of the Polish Mathematical Society of Warsaw Division (1957-58, 1963-64).
- Victor W. Marek received his M.A. and Ph.D. degrees from the University of Warsaw, Poland. He is currently a Professor of Computer Science at the University of Kentucky, Lexington. He is an author of over 80 papers in the areas of mathematical logic, computer science and artificial intelligence, including three books.