Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

L. Kossen

Resolution and logical consequences

Computer Science/Department of Software Technology Report CS-R9023 May

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Resolution and Logical Consequences

Louis Kossen
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract

Completeness of binary resolution is defined in refutation form: if a set of clauses in first order logic is inconsistent, then the empty clause (□) can be deduced by binary resolution. In this paper we formulate and prove an extended completeness theorem for binary resolution. It will be proved that for each clause which is a non-trivial logical consequence of a theory, a clause subsuming that clause can be generated by binary resolution. An extended completeness theorem will also be formulated and proved for P_1 resolution. Furthermore we will show how extended completeness can be applied to knowledge based systems. According to refutation completeness, resolution can be used to deduce a fact by refuting its negation. These refutation proofs are hard to understand. We will show, by extended completeness, that in the propositional case resolution can be used to generate a set of so-called non-trivial minimal logical consequences together with an explanation of these consequences. This set contains also the facts which can be verified by means of refutation completeness. Our approach contributes to better explanation facilities of knowledge based systems.
Keywords & Phrases: theorem proving, resolution, knowledge representation, knowledge based systems.
1985 Mathematics Subject Classification: 68Q40, 68T15, 68T30
1987 Computing Reviews Categories: F.4.1, I.2.2, I.2.4.
Note: The work in this document was conducted as part of the PRISMA project, a joint effort with Philips Research Eindhoven partially supported by the Dutch "Stimulerings-projectteam Informatica-onderzoek" (SPIN).

1 Introduction

In this paper we will formulate and prove an extended version of completeness of resolution. Completeness of resolution is defined in a refutation form i.e. if a theory is inconsistent then it is possible to generate the empty clause by resolution. However, the usual definition of completeness for the various inference systems is not the same. For example, the definition of completeness for natural deduction is stronger than the definition of completeness for resolution: if a sentence is a logical consequence of a first order logic theory then it can be deduced from the theory by natural deduction. Refutation completeness captures only the case in which a theory is inconsistent. We wondered whether the completeness results for resolution could also be extended to consistent theories i.e. which clauses can be generated by resolution from a consistent theory.

It is not possible to generate all logical consequences of a theory by resolution but we will prove that for each non-trivial logical consequence, i.e. a logical consequence which is not a tautology, a clause subsuming that logical consequence can be generated. We shall show that the subsumption relation defined on first order clauses is not well-founded i.e. there are theories containing infinite sequences of clauses in which each clause is properly subsumed by its successor. However, the subsumption relation defined on ground clauses, clauses which do not contain variables, induces a well-founded ordering. Hence, it makes sense to reason about minimal (w.r.t. the subsumption ordering) logical consequences of a ground theory. This and the extended completeness definition implies that each non-trivial minimal logical consequence of a ground theory can be generated. Because knowledge bases in expert systems can be considered as ground theories we have a direct application of the extended completeness theorem to expert systems. Moreover, the construction of the proof presented in this paper implies that for each non-trivial minimal logical consequence there exists a direct deduction which can serve as an explanation.

The facts presented in this paper are results from research in the area of the expert systems. Initially, the problem was how to test the consistency of a knowledge base in a logic formalism and, if consistent, how to generate a model. In [B] an algorithm is presented which tests consistency of a knowledge base in a logic formalism and which generates a model if the theory is consistent. A disadvantage of an arbitrary model is that if the knowledge base is extended with new information the model may not be valid any more. We will show that the set of non-trivial minimal logical consequences is the most complete and accurate information that can be generated from a knowledge base. If the knowledge base is extended with new information the clauses in the set of minimal logical consequences may not be minimal anymore but the clauses still hold.

For definitions and facts concerning resolution we rely on [CL,L,R1,R2] and for the more logic related subjects we refer to [D] . In the first section we will give a short overview of the most important theorems and definitions of the theory of resolution. The extended completeness of binary resolution and P_1 resolution is stated and proved in the third section. In the fourth section we will take a closer look at subsumption and its relation to first order logic theories and ground theories. In this section we will introduce the notion minimal logical consequence for ground clauses. The relevance of extended completeness for expert systems will be discussed in the fifth section. A conclusion, a few remarks and the references can be found in the last two sections.

2 Preliminaries

In this section we will recall the most crucial and important definitions and theorems concerning resolution. We assume the reader is familiar with the basic definitions of resolution, for which we rely on [R1] and [L]. In the literature on resolution clauses are denoted either by sets of literals or disjunctions of literals. In this paper we will consider clauses as sets of literals. The empty clause is the clause which contains no literals at all. This clause is denoted as $\Box$. We assume that clauses are equal modulo a renaming of the variables. We will start this section with the introduction of notational convenience which will be used throughout this paper. The first definition concerns the subsumption relation between clauses which will play an important role in this paper.

Definition 2.1 *A clause C* subsumes *a clause C' if there is a substitution φ such that all the literals of $C\varphi$ are in C'. This is denoted as $C \preceq C'$.*

We will make a distinction between trivial logical consequences and non-trivial logical consequences. The trivial logical consequences are the tautologies. The tautologies can be characterized as follows:

Fact 2.2 *A clause is a tautology if and only if it contains two complementary literals.*

In the next definition the binary resolution is defined. In [R1] the set notation is used and for that reason we have relied on the definition of binary resolution presented in [R2]. In this definition the complement of a literal L is denoted $\overline{L}$. A clause C from which a literal L is deleted, is denoted $C \setminus \{L\}$. We assume the reader is familiar with the notions unifier and most general unifier. Definitions of these notions can be found in [L,R1].

Definition 2.3 *Let C_1 and C_2 be two clauses having no variables in common and containing the literals $\{L_{11}, ..., L_{1n}\} \subseteq C_1$ $(n > 0)$ and $\{L_{21}, ..., L_{2m}\} \subseteq C_2$ $(m > 0)$ respectively. If there is a most general unifier φ of $\{L_{11}, ..., L_{1n}, \overline{L}_{21}, ..., \overline{L}_{2m}\}$ then* a binary resolvent *of C_1 and C_2 is the clause $(C_1\varphi \setminus \{L_{11}\varphi\}) \cup (C_2\varphi \setminus \{L_{21}\varphi\})$.*

In the introduction we mentioned the fact that we will not only prove an extended completeness theorem for binary resolution but also for P_1 resolution. This form of resolution is introduced in [R2]. Hyperresolution is the successive application of a number of P_1 resolution steps. The definition of P_1 resolution can be found below.

Definition 2.4 *A clause C is a* P_1 resolvent *of clauses C_1 and C_2 if C is a binary resolvent of C_1 and C_2 and one of the clauses C_1 and C_2 is a positive clause i.e. contains just positive literals.*

The facts and definitions we present at the end of this section also hold for P_1 resolution. In the next definition we introduce notation for more than one binary resolution step.

Definition 2.5 *If T is a theory then a* br-deduction (P_1-deduction) *from T is a finite sequence of of clauses $[C_1, ..., C_n]$ satisfying the property that each C_i is the binary (P_1) resolvent of two clauses from the set $T \cup \{C_l : 1 \leq l < i\}$*

By this definition we come to the following definition of the derivability of a clause from a theory.

Definition 2.6 *If T is theory then $T \vdash_{br} C$ ($T \vdash_{P_1} C$) if C is in T or there is a br-deduction (P_1-deduction) $[C_1, ..., C_n]$ with $C = C_n$ from T.*

Two relevant properties of an inference rule are completeness and soundness. In the introduction we already mentioned the difference in definitions of completeness for the various inference systems. Completeness always indicates what kind of formulas can be generated from a theory by means of that inference rule. An inference rule is called sound if the rule respects all interpretations i.e. if an interpretation satisfies a theory then it satisfies also the inferred formulas. In the following fact we will formalize these notions for binary resolution and P_1 resolution. By $T \models C$ we denote that C is a logical consequence of T.

Fact 2.7 *The completeness and soundness definitions for binary and P_1 resolution are stated as follows:*

1. (soundness) *If T is a theory and C is a clause then the following holds:*

$$T \vdash_{br} C \Rightarrow T \models C$$

$$T \vdash_{P_1} C \Rightarrow T \models C$$

2. (refutation completeness) *If T is a theory then the following holds:*

$$T \textit{ is inconsistent} \Rightarrow T \vdash_{br} \Box$$

$$T \textit{ is inconsistent} \Rightarrow T \vdash_{P_1} \Box$$

We finish this section with two facts which will play an important role in the proof of extended completeness. The proofs of these facts can be found in [L,R1].

Fact 2.8 (Herbrand's Theorem) *A theory T is inconsistent if and only if there is a finite inconsistent set of ground instances of clauses in T.*

Fact 2.9 (Lifting Lemma) *Let C_1' and C_2' be instances of clauses C_1 and C_2 respectively. If C' is a binary resolvent of C_1' and C_2' there is a binary resolvent C of C_1 and C_2 such that C' is an instance of C.*

3 Resolution and logical consequences.

In this section we will formulate and prove an extended completeness theorem for both binary resolution and P_1 resolution. The proof we present in this section is based on refutation completeness and Herbrand's theorem. We will prove the extended completeness theorem for ground clauses and extend it to first order clauses later on. First we prepare by some lemmas of which we omit the straightforward proofs.

Lemma 3.1 *S is a finite set of ground clauses, S contains a unit clause $U = \{L\}$ and S' is obtained from S by deleting all the clauses containing literal L and deleting the literal $\overline{L}$ from all clauses containing $\overline{L}$. Then, S' is consistent if and only if S is consistent.*

The constructive part of the proof can be found in the proof of the next lemma.

Lemma 3.2 *If T is a consistent theory, C a ground clause and C is a non-trivial logical consequence of T then there exists a clause C' (possibly containing variables) such that $T \vdash_{br} C'$ and $C' \preceq C$.*

Proof. Let C be the clause:

$$C = \{A_1, ..., A_p, \neg B_1, ..., \neg B_q\}$$

in which the literals A_i, B_j $(1 \leq i \leq p, 1 \leq j \leq q)$ are ground.

If there is a clause $C' \in T$ and $C' \preceq C$ we are done. What is left to prove is that if there is no clause $C' \in T$ such that $C' \preceq C$ then there is a br-deduction $[C_1, ..., C_n]$ from T such that $C_n \preceq C$.

Globally, the proof can be split up in two parts. First we will show that a subclause of C, that is a clause containing just literals from C, can be derived from a finite set of closed instances of clauses from T. Secondly, to complete the proof, we lift this br-deduction from this set of closed instances to a br-deduction from T. The clause generated by this br-deduction satisfies the conditions stated in the theorem.

The set $\overline{C}$ is the set of the negation of literals from C.

$$\overline{C} = \{\neg A_1, ..., \neg A_p, B_1, ..., B_q\}$$

We also have to consider the unit clauses which contain a literal from C and $\overline{C}$. These sets will be denoted U_C and $U_{\overline{C}}$

$$U_C = \{\{A_1\}, ..., \{A_p\}, \{\neg B_1\}, ..., \{\neg B_q\}\}$$

$$U_{\overline{C}} = \{\{\neg A_1\}, ..., \{\neg A_p\}, \{B_1\}, ..., \{B_q\}\}$$

Because $T \models C$, the set $T' = T \cup U_{\overline{C}}$ is inconsistent. According to Herbrand's Theorem there is a finite set S of closed instances of clauses from T' which is inconsistent. We will show there is a br-deduction of a subclause of C from S which consists just of ground instances of clauses from T.

The set T is consistent and the set S is inconsistent, so according to Herbrand's Theorem S contains at least one element from the set $U_{\overline{C}}$. Furthermore, because C is not a tautology the set S contains ground instances of clauses from T. So, we have:

$$S \cap U_{\overline{C}} = \{\{L_1\}, ..., \{L_l\}\} \; (l \geq 1)$$

We will reduce the set S to a set S' by successively applying the procedure described in Lemma 3.1 with elements from $U_{\overline{C}}$. The set S' is obtained from S by constructing a sequence of sets

$$S \equiv S_0, ..., S_l \equiv S'$$

where S_{i+1} is obtained from S_i as follows:

- delete all clauses containing literal L_{i+1}
- delete literals $\overline{L_{i+1}}$ from all the clauses containing literal $\overline{L_{i+1}}$

Because C is not a tautology, this set is uniquely determined and independent from in which order the elements of $S \cap U_{\overline{C}}$ are processed. In a formal way S' can be stated as:

$$S' = \{C'' : C'' \text{ is ground} \wedge C'' \cap C = \emptyset \wedge C'' \cap \overline{C} = \emptyset \wedge \exists C' \in S(C'' \preceq C' \wedge C' \setminus C'' \subseteq C)\}$$

By successively applying Lemma 3.1 we verify that the inconsistency of S is propagated to S'. Observe that due to the transformation we have $S' \cap U_{\overline{C}} = \emptyset$. Because we assumed that C is not subsumed by a clause from T, the set S' does not contain the empty clause. Hence, for each clause in S' there is a clause in S which is a ground instance of a clause from T and the difference between the clauses are literals from $\{\overline{L_1}, ..., \overline{L_l}\} \subseteq C$.

According to refutation completeness (Fact 2.7) there is a a br-deduction $[C'_1, ..., C'_n]$ from S' where $C'_n = \Box$. The br-deduction $[C'_1, ..., C'_n]$ from S' will be transformed into a br-deduction $[C_1, ..., C_n]$ from $S \setminus U_{\overline{C}}$ such that $C_n \preceq C$. Observe that each clause in S' is obtained from a clause in S by deletion of a number (possibly 0) literals from C. By induction on the length of $[C'_1, ..., C'_n]$ we will define this transformation where in each step C'_i is transformed into C_i. Simultaneously we prove by induction the following: $C'_i \preceq C_i$ and $C_i \setminus C'_i \subseteq C$. The transformation is defined as follows:

$i = 1$: C'_1 is the binary resolvent of two clauses from S'. Choose a pair of parents. Because all clauses from $U_{\overline{C}}$ have been deleted, each of these parent clauses has at least one corresponding clause in $S \setminus U_{\overline{C}}$. Now let C_1 be the binary resolvent of two corresponding clauses of the parent clauses from $L \setminus U_{\overline{C}}$. The corresponding clauses of the parents of C'_1 differs just from literals in C. We conclude that $C_1 \setminus C'_1 \subseteq C$.

$i > 1$: C'_i is the binary resolvent of two clauses from $S' \cup \{C'_j : 1 \leq j < i\}$. Choose a pair of parent clauses in $S' \cup \{C'_j : 1 \leq j < i\}$. If a parent clause is in S' then choose a corresponding clauses in $S \setminus U_{\overline{C}}$. If a parent is in $\{C'_j : 1 \leq j < i\}$ then take for the new parent clause the corresponding extended version in $\{C_j : 1 \leq j < i\}$. Now let C_i be the binary resolvent of these new parent clauses which resolve upon the same literal as the parents of C'_i. If one of the parent clauses of C'_i is in S' then the corresponding clause in S differs in literals from C. If a parent clause is in $\{C'_j : 1 \leq j < i\}$ then, using the induction hypothesis, the corresponding clause in $\{C_j : 1 \leq j < i\}$ differs also in literals from L_C. Conclusively, the difference between C_i and C'_i are literals from C.

Following the inductive definition of the transformation the next two observations are checked:

1. $[C_1, ..., C_n]$ is a br-deduction from $S \setminus U_{\overline{C}}$
2. C_i differs from C'_i $(1 \leq i \leq n)$ in a finite (possibly 0) number of literals from C

We will prove that C_n has a non empty extension with literals from C. If $C_n = C'_n = \Box$ then the first observation and Herbrand's Theorem would imply the inconsistency of T. The theory T is consistent, so the clause C_n is a non empty extension of $\Box$ and according to the second observation we conclude that $C_n \preceq C$. This concludes the first part of the proof.

The last part of the proof is lifting the br-deduction $[C_1, ..., C_n]$ from $S \setminus U_{\overline{C}}$ to a br-deduction $[C''_1, ..., C''_n]$ from T where $C''_n \preceq C$. The parents of C_1 are instances of clauses from T. By the Lifting Lemma we can lift the deduction of C_1 into a deduction of C''_1 such that the parents of C_1 are instances of the parents of C''_1 and C_1 is an instance of C''_1. We can successively apply the Lifting Lemma on each C_i of the br-deduction $[C_1, ..., C_n]$ from $S \setminus U_{\overline{C}}$ and we obtain a br-deduction $[C''_1, ..., C''_n]$ from T such that $C''_n \preceq C$. ■

Lemma 3.3 *Let T be a theory and C a clause, with variables $x_1, ..., x_p$. If $c_1, ..., c_p$ are constants not in the clauses of T and C_g is the clause C in which all x_i have been substituted by c_i then:*

$$T \models C \Rightarrow T \models C_g$$

Proposition 3.4 *If T is a consistent theory and C a non-trivial logical consequence then there is a clause C' such that $T \vdash_{br} C'$ and $C' \preceq C$.*

Proof. We have already proved this theorem for ground clauses. Let C_g be the ground clause defined in the same way as in the previous lemma. Because C_g is ground there is a br-deduction $[C_1, ..., C_n]$ from T where $C_n \preceq C_g$. Because the unifiers used in a binary resolution step are most general and the constants c_i $(1 \leq i \leq p)$ do not occur in T we conclude that the clauses in the deduction $[C_1, ..., C_n]$ do not contain any of the constants c_i $(1 \leq i \leq p)$. Hence, the clause C_n does not contain any constants c_i $(1 \leq i \leq p)$ and the substitution belonging to the subsumption of C_g by C_n has the following form:

$$\varphi = \{y_1/t_1(c_1, ..., c_p), ..., y_m/t_m(c_1, ..., c_p)\}$$

where $t_i[c_1, ..., c_p]$ is a closed term. Without loss of generality we assume that the variables of C are not used in the deduction then what follows is indeed a substitution φ':

$$\varphi' = \{y_1/t_1(x_1, ..., x_p), ..., y_m/t_m(x_1, ..., x_p)\}$$

and this substitution satisfies $C'\varphi' \preceq C$. ∎

Theorem 3.5 (extended completeness) *If T is a theory and C is a non-trivial logical consequence of T, then there exists a clause C' such that $T \vdash_{br} C'$ and $C' \preceq C$.*

Proof. We distinguish two cases:

1. T is inconsistent. According to refutation completeness the empty clause can be generated and the empty clause subsumes C.
2. T is consistent. Proposition 3.4 implies the existence of a C' such that $T \vdash_{br} C'$ and $C' \preceq C$. ∎

We finish this section with the extended completeness theorem for P_1 resolution. Although there is a difference between binary resolution and P_1 resolution the definition of refutation completeness is the same for both inference rules. However, the extended completeness definition for P_1-resolution is different from extended completeness definition of binary resolution. The proof proceeds almost in the same way and therefore we will only indicate at which points the proof for P_1-resolution deviates from the previous proof. First of all we will introduce an alternative version of Lemma 3.2.

Lemma 3.6 *If T is a consistent theory, C a positive ground clause which is a non-trivial logical consequence, then there exists a clause C' (possibly containing variables) such that $C' \preceq C$ and $T \vdash_{P_1} C'$.*

Proof. The proof of this lemma proceeds almost in the same way as for binary resolution. If C is the clause $C = \{A_1, ..., A_p\}$ then the clauses C and $\overline{C}$ are

$$C = \{A_1, ..., A_p\}$$

$$\overline{C} = \{\neg A_1, ..., \neg A_p\}$$

And the corresponding set of unitclauses U and $U_{\overline{C}}$ are

$$U_C = \{\{A_1\}, ..., \{A_p\}, \{\neg B_1\}, ..., \{\neg B_q\}\}$$

$$U_{\overline{C}} = \{\{\neg A_1\}, ..., \{\neg A_p\}, \{B_1\}, ..., \{B_q\}\}$$

There is a set S containing instances of clauses from $L_{\overline{C}}$ and instances of clauses from T. The set S is transformed into a set S' in the same way as in the proof of Lemma 3.2. There is a P_1 deduction $[C'_1, ..., C'_n]$ from S' where $C'_n = \Box$. Similar to the proof of Lemma 3.2 we will lift this P_1-deduction $[C'_1, ..., C'_n]$ from S' to a P_1-deduction $[C_1, ..., C_n]$ from $S \setminus U_{\overline{C}}$. From the proof of Lemma 3.2 we have that $[C_1, ..., C_n]$ is a br-deduction from $S \setminus U_{\overline{C}}$. So, what is left is to verify that all binary resolution steps in the deduction $[C_1, ..., C_n]$ are P_1-resolution steps. One of the parents of all C'_i $(1 \leq i \leq n)$ is a positive clause. The corresponding clause in $S \cup \{C_j : 1 \leq j < i\}$ differs from the original parents in literals from C. Hence, we conclude that the corresponding clause of the parent clause which is positive clause is still a positive clause. Conclusively, a P_1 step remains a P_1 step. The remaining part of the proof proceeds in the same way as in the proof of Lemma 3.2 ■

Theorem 3.7 (extended completeness for P_1 resolution) *If T is a theory, C is positive clause and C is a non-trivial logical consequence of T then there is a clause C' such that $T \vdash_{P_1} C'$ and $C' \preceq C$.*

Proof. It is easy to verify that Proposition 3.4 also holds for P_1 resolution. Then in the same way as in Theorem 3.5 we prove Theorem 3.7. ■

4 Subsumption

In the extended completeness definition we proved in the previous section the subsumption relation plays an important role. Therefore we will take a closer look at subsumption. We will indicate what kind of relation subsumption induces on ground clauses and clauses containing variables. First of all we introduce the definitions of two kinds of relations.

Definition 4.1 *A relation $\leq$ on S is a* partial order *if:*

1. $\forall a, b, c \in S \; a \leq b \wedge b \leq c \Rightarrow a \leq c$ *(transitivity)*
2. $\forall a, b \in S \; a \leq b \wedge b \leq a \Rightarrow a = b$ *(anti-symmetry)*

If $a \leq b$ and $a \neq b$ the we also write $a < b$. The second kind of relation we will consider is a refinement of a partial order.

Definition 4.2 *A partial order $\leq$ on a set S is* well-founded *if there are no infinite descending sequences*

$$... a_n < a_{n-1} < ... < a_2 < a_1$$

We will make a distinction between subsumption and proper subsumption.

Definition 4.3 *A clause C* properly subsumes *a clause C', notated as $C \prec C'$, if $C \preceq C'$ and not $C' \preceq C$.*

Firstly, we consider the relation between ground clauses and subsumption. Subsumption restricted to ground clauses is equivalent to the subset relation between ground clauses.

Definition 4.4 *If C, C' are two clauses and the literals of C' are also in C then C' is a* subclause *of C. The clause C is a* proper subclause *of C' if C' is not a subclause of C.*

With this definition in mind it can easily be verified that the (proper) subsumption induces a partial order on ground clauses i.e. (proper) subsumption is transitive and anti-symmetric. Because each clause contains a finite number of literals it is not possible to construct an infinite strictly descending sequence of clauses. Hence, we conclude that the subsumption relation induces a well-founded partial order on the ground clauses. Therefore, it makes sense to reason about minimal elements.

Definition 4.5 *If T is a ground theory and C a ground clause then C is a minimal logical consequence of T if C is a logical consequence and for all other logical consequences C' of T we have $C' \not\preceq C$.*

In the next proposition extended completeness and the notion of minimal logical consequence have been combined.

Proposition 4.6 *If T is a ground theory and C a non-trivial minimal logical consequence of T, then $T \vdash_{br} C$.*

Proof. Definition 4.5 and extended completeness of binary resolution. ■

The proper subsumption relation defines also a partial order on first order clauses. The proper subsumption clearly satisfies the transitivity. Anti-symmetry of proper subsumption is explicitly included in the definition of proper subsumption. This is not the case for the ordinary subsumption relation where anti-symmetry is not satisfied. For example, the clauses $\{P(x)\}$ and $\{P(z), P(y)\}$ are subsuming each other but they are not equal in the way we mentioned in section two i.e. they are equal modulo a variable renaming.

The proper subsumption relation induces a partial order on the first order clauses but this partial order is not well-founded. The sequence shown below is infinite and strictly descending. Each clause is properly subsumed by its successor.

$$C_1 = \{P(x, x)\}$$

$$C_2 = \{P(x, y_1), P(y_1, x)\}$$

$$C_3 = \{P(x, y_1), P(y_1, y_2), P(y_2, y_3), P(y_3, x)\}$$

$$C_4 = \{P(x, y_1), P(y_1, y_2), P(y_2, y_3), P(y_3, y_4), P(y_4, y_5), P(y_5, y_6), P(y_6, y_7), P(y_7, x)\}$$

$$\vdots$$

So, although the subsumption relation on first order clauses is not well-founded we can apply the results for ground clauses to knowledge based systems because knowledge bases can be considered as ground theories. In the next section we will discuss the application of extended completeness to knowledge based systems.

5 Application of extended completeness

In this section we will show, by means of an example, how resolution can be applied in knowledge based systems. In the first section we claimed that the set of minimal logical consequences is the most accurate and complete information that can be retrieved from a knowledge base. This set is to be preferred above an arbitrary model because its truth is preserved after addition of new consistent information. Secondly, for each non-trivial minimal logical consequence exists a direct deduction which can serve as an explanation.

In this section we will show that there are two more interesting aspects about the set of non-trivial minimal logical consequences. We will show that the set of minimal logical consequeces can be used to detect the incomplete information in the rule base part of the knowledge base. Another interesting aspect is that the set of non-trivial minimal logical consequences contains the facts that can be verified due to refutation completeness. Mostly resolution is used in knowledge based systems is to support the question-answer facility of the system. Given a knowledge base K the fact F can be verified by applying resolution on the set $K \cup \{\{\neg F\}\}$. If the empty clause can be generated from this set the fact F is a logical consequence of K. However, it may not be clear at all what kind of question should be asked. If the question can be confirmed then the fact also appears in the set of minimal logical consequences. Another possibility is that a single fact cannot be verified but that it is still possible to retrieve some information. These minimal dependencies can be found in the set of non-trivial minimal logical consequences.

The knowledge base we will use in this section is shown in table 5.1. The rule base part consists of four rules and one fact specific for a particular vehicle has been added. The rule base part of the knowledge base contains general information on the subject "vehicles". The given fact F_1 is information specific for a particular vehicle.

$$
\begin{aligned}
R_1 &= \{\ \neg\text{four\_wheel(vehicle)},\ \text{car(vehicle)}\ \}\\
R_2 &= \{\ \neg\text{two\_wheel(vehicle)},\ \text{bike(vehicle)},\ \text{motorcycle(vehicle)}\ \}\\
R_3 &= \{\ \neg\text{engine(vehicle)},\ \text{car(vehicle)},\ \text{motorcycle(vehicle)}\ \}\\
R_4 &= \{\ \text{engine(vehicle)},\ \text{bike(vehicle)}\ \}\\
F_1 &= \{\ \text{two\_wheel(vehicle)}\ \}
\end{aligned}
$$

table 5.1. the knowledge base.

First of all we will show that the rule base part has some defects. If the knowledge base has some undesired models then we can conclude that the rule base part of the knowledge base is incomplete.

$$
\begin{aligned}
M_1 &= \{\ \neg\text{four\_wheel(vehicle)},\ \text{car(vehicle)}\ \}\\
M_2 &= \{\ \text{bike(vehicle)},\ \text{motorcycle(vehicle)}\ \}\\
M_3 &= \{\ \neg\text{engine(vehicle)},\ \text{car(vehicle)},\ \text{motorcycle(vehicle)}\ \}\\
M_4 &= \{\ \text{engine(vehicle)},\ \text{bike(vehicle)}\ \}\\
M_5 &= \{\ \text{two\_wheel(vehicle)}\ \}
\end{aligned}
$$

table 5.2. non-trivial minimal logical consequences of knowledge base in table 5.1.

The set of non-trivial minimal logical consequences can be used to find the allowed models more easily. In table 5.2 we have shown the non-trivial minimal logical consequences of the knowledge base of table 5.1. From the clauses in table 5.2 we can conclude that there exist some undesired models. For example it is possible that a vehicle is both a car and a bike. To repair these shortcomings of the rule base we have to add some new rules. The rules which have to be added are shown in table 5.3. Notice, that the rules in table 5.3. guarantee that a vehicle can be of just one type.

$$R_5 = \{ \neg\text{four\_wheel(vehicle)}, \text{car(vehicle)} \}$$
$$R_6 = \{ \neg\text{car(vehicle)}, \neg\text{motorcycle(vehicle)} \}$$
$$R_7 = \{ \neg\text{bike(vehicle)}, \neg\text{motorcycle(vehicle)} \}$$

table 5.3. extension of knowledge base in table 5.1.

Although we have completed the knowledge base it is still not possible to determine with what the type of the vehicle is by one single question. However, it is possible to draw some conclusions from the knowledge bases in table 5.1 and 5.3. These conclusions can be found in the set of non-trivial minimal logical consequences of the extended knowledge base. This set is listed in table 5.4.

$$M_1 = \{ \neg\text{car(vehicle)} \}$$
$$M_2 = \{ \text{bike(vehicle)}, \text{motorcycle(vehicle)} \}$$
$$M_3 = \{ \neg\ \text{bike(vehicle)}, \neg\text{motorcycle(vehicle)} \}$$
$$M_4 = \{ \neg\text{engine(vehicle)}, \text{motorcycle(vehicle)} \}$$
$$M_5 = \{ \neg\text{engine(vehicle)}, \neg\text{bike(vehicle)} \}$$
$$M_6 = \{ \text{engine(vehicle)}, \text{bike(vehicle)} \}$$
$$M_7 = \{ \text{two\_wheel(vehicle)} \}$$
$$M_8 = \{ \neg\text{four\_wheel(vehicle)} \}$$

table 5.4. non-trivial minimal logical consequences of extended knowledge base

From the clauses in table 5.4 we can conclude that the vehicle is not a car but a bike or a motorcycle. Notice, that the unicity of the vehicle is a consequence of M_3. These observations could not have been made with one single question.

For each clause in the set of minimal logical consequences exists a direct deduction which can serve as an explanation. For example, the consequence M_2 in table 5.4 can be deduced with a resolution step of clauses R_2 and F_1. Another example is the minimal logical consequence M_1 which can be obtained from the clauses R_2, R_5, R_6 and F_1. The shortest deduction can be used as an explanation of the consequences.

6 Conclusions and remarks

In this report we have proved an extended completeness theorem for both binary and P_1 resolution. In the ground case we are able to generate a set of minimal logical consequences. We think that the minimal logical consequences are the most exact and accurate information one can retrieve from a knowledge base. There is a direct deduction for each minimal logical consequence which can serve as an explanation for that consequence. We also indicated that the set of minimal logical consequences can also be used to reveal deficiencies of the rule base part of the knowledge base.

A very important role in the research presented here was played by a resolution based theorem prover called ITP (Interactive Theorem Prover). In this theorem prover, build by William McCune of the Argonne National Laboratory, various inference strategies have been implemented. A description of these inference strategies can be found in [W]. Among the implemented inference strategies are binary resolution, hyperresolution and subsumption. Analysis of the output for several knowledge bases gave us some idea about what kind of clauses are generated. The combination of binary resolution and subsumption is a procedure which terminates and generates the minimal logical consequences of a ground theory.

Finally, I want to thank Marc Bezem for careful proofreading this paper and his useful remarks.

References

[B] M. Bezem, *Consistency of rule-based experts systems.* Proceedings of the 9-th Conference on Automated Deduction, Springer Lecture Notes in Computer Science 310, pp. 151-161, Springer Verlag, Berlin 1987.

[CL] C. L. Chang, R. C. T. Lee, *Symbolic logic and mechanical theorem proving.* Academic Press, New York 1973.

[D] D. van Dalen, *Logic and structure.* 2nd edition, Springer-Verlag, Berlin 1983.

[L] D. W. Loveland, *Automated Theorem Proving: a Logical Basis.* North-Holland, Inc, New York 1982.

[R1] J. A. Robinson, *A machine oriented logic based on the resolution principle.* Journal of the ACM 12, 1, pp. 23-41, 1965.

[R2] J. A. Robinson, *Automatic deduction with hyper-resolution.* International Journal of Computer Mathematics 1, pp. 227-234, 1965.

[W] L. Wos, R. Overbeek, E. Lusk and J. Doyle, *Automated Reasoning: Introduction and applications.* Prentice Hall, 1984.