A Note on Backward Dual Resolution and Its Application to Proving Completeness of Rule-Based Systems*

Antoni Liqeza Institute of Automatics AGH al. Mickiewicza 30, 30-059 Krakow, Poland

tel. &; fax: (48 12) 34 15 68, e-mail: ali@earth.ia.agh.edu.pl

Abstract

In this paper, a method of theorem proving dual to resolution method is presented in brief. The investigated method is called backward dual resolution or bd-re solution, for short. The main idea of bd-resolution consists in proving validity of a formula in disjunctive normal form, by generating an empty tautology formula from it; it is shown that the initial formula is a logical consequence of the obtained tautology. An idea of the theorem proving method is outlined, and its application to checking completeness of rulebased systems is investigated. A formal definition of completeness and specific completeness are stated and an algorithm for completeness verification is proposed. Moreover, a generalized bd-resolution aimed at proving completeness under intended interpretation is defined.

Introduction

Resolution theorem proving has gained a great popularity after it was first described by Robinson [1965]. The method combines in a single rule the power of other rules. and, due to its uniformity, can be easily implemented for automated theorem proving with computers.

In this paper, an idea of a theorem proving method dual to classical resolution is investigated in brief. The method can be related to the inverse method of Maslov [Maslov, 1964] (see also [Maslov, 1968] for the most complete presentation), and according to [Kuehner, 1971] it can be derived from the inverse method; in fact, it falls into the class of localized inverse methods [Kuehner, 1971; Maslov, 1971]. However, it seems most straightforward to introduce the investigated method as one dual to classical resolution.

The presented method is called backward dual resolution or bd-resolution, for short. The proposed method

*This research was supported by State Committee on Scientific Research within the Project Computer Systems of Control and Decision Making - Elements of Theory, Formal Tools and Computer Aided Design, Grant No. 8-8528-9102; a first version of this paper appeared as a Technical Report of Laboratoire d'Automatique et d'Analyse des Systemes (LAAS du CNRS) No. 92214, Toulouse, France, 1992

is analogous to resolution method. Instead of checking unsatisfiability of a set of clauses we rather try to prove validity of a given formula in disjunctive normal form. Thus, the initial form is in fact dual to the one in resolution method. Further, the proposed method works in fact backwards. This means that during the process of derivation one generates new formulae from parent formulae starting from the initial formula to be proved - but with regard to logical inference, the alternative of parent formulae is a logical consequence of the derived formula! Thus, at any step of reasoning the derivation process is reversed with regard to finding logical consequences. The process of derivation is successful if it eventually ends up with an empty tautology formula which (here) is always true - in this case, the initial formula, as a logical consequence of it, is proved. Since the method is dual to resolution, all but necessary technical details and proofs are omitted here; one can find them in [Lige.za, 1992b].1

The motivation for bd-resolution follows from logical investigation of rule-based control systems. Throughout this paper, by a rule-based control system (knowledgebased control system) we shall understand a system designed to control some object and consisting of a set of rules defining the possible control actions. A rule-based control system is assumed to operate according to the following scheme: the current state of the controlled system is observed, then a rule with satisfied preconditions is selected, and the specified by this rule control action is executed; the basic cycle is repeated in an "endless loop" until it is stopped (either by some of the rules or as a result of some external interrupt). Such systems were discussed in [Tzafestas and Lige/za, 1989] and more

thoroughly in [Lige,za, 1992b].

BD-resolution was initially introduced as a basic reasoning paradigm for knowledge-based control systems [Ligeza, 1992a; Ligeza, 1992b]. There are at least three standard issues concerning logical investigation of such systems; these are: (I) checking if logical description of the current situation satisfies the formula defining preconditions of a rule (i.e. checking if a rule can be fired), (II) checking if some two formulae defining preconditions of two rules may be simultaneously satisfied (i.e. check-

¹A copy of this report should be available from Documentation Service, L.A.A.S. du C.N.R.S., 7, Av. du Colonel-Roche, 31-077 Toulouse Cedex, France.

ing if some two rules can succeed for the same state in such a case the system is indeterministic and there is a need for conflict resolution), and (III) checking if the alternative of precondition formulae of all the rules constitutes a tautology (i.e. checking if the control system can react in any state an so it is complete). The above tasks can be approached with use of different inference mechanisms (including the classical resolution), but it seems that bd-resolution fits best the purpose. The advantage of bd-resolution consist in the possibility of direct reasoning with formulae describing situations and preconditions in a most natural way. Further, in case of incomplete systems formulae defining specific completeness are directly generated. Here we investigate only the possible application of bd-resolution concerning completeness verification of rule-based systems (problem (III)). The other problems and related issues are presented in some details in [Ligeza, 1992b].

The problem of logical checking of completeness, consistency, and related issues seems to be of both theoretical interest and practical importance, but simultaneously largely ignored in the literature. A comprehensive recent review providing a recapitulation of the problems and summarizing current results is given in [Andert, 1992]; almost complete list of problem literature can be found there as well

With regard to completeness of rule-based systems two basic approaches can be noticed. The most popular one consists in, roughly speaking, an exhaustive enumeration of possible input data and systematic inspection of a given set of rules versus a table containing all possible parameters and conditions combinations. This kind of approach can be called an exhaustive completeness check [Andert, 1992]. Some examples of this approach are presented in [Suwa et al, 1984; Cragun and Steudel, 1987]. A basic approach ([Suwa et al, 1984]) consists in finding all parameters used in conditions of rules, generating a table displaying all possible combinations of parameter values, and checking the set of rules against the table so as to verify completeness and detect missing rules. This kind of approach can be also referred to as static verification of the set of rules.

The other approach consists in a run-time validation and verification of the expert system with use of selected set of test cases [Tepandi, 1990]. The selected example problems should also provide an exhaustive list of possible cases. Some other approaches of this kind are also discussed in [Andert, 1992]. Methods of this sort can be referred to as *dynamic* verification of rule-based system.

In this paper another logic-based approach is proposed. The approach neither requires the generation and use of tables containing possible condition/parameter combinations, nor it is based on exhaustive test of possible cases. Note that, in case of use of full first-order logic for building preconditions of the system rules an exhaustive check would seldom be feasible, since even for a language containing only one function symbol the Herbrand universe ([Chang and Lee, 1973]) is infinite.

The proposed approach can be applied directly to systems in which rules are constructed with use of predicate logic notation such that conjunction is the basic connec-

tor between facts (conditions); more complex formulae are built with use alternative. The negation is to be expressed explicitly, i.e. facts of the form p and -p should be used. The method can be used for rules constituting a "single-layer" of reasoning, i.e. during reasoning one of the rules from a considered set is to be selected and fired; thus, the proposed approach is especially convenient for knowledge-based control systems [Ligeza, 1992b;

Tzafestas and Lige,za, 1989].

2 BD-Resolution

Let us briefly recapitulate the main idea of resolution theorem proving (the resolution method is presented in e.g. [Chang and Lee, 1973; Genesereth and Nilsson, 1987]). Throughout the paper we shall made use of some commonly used logical symbols, including $\models (\models_I)$ for denoting logical satisfaction (satisfaction under interpretation I) and \vdash for provability ([Genesereth and Nilsson, 1987]). An empty tautology formula (always true!) will be denoted with T.

Without going into details, resolution theorem proving proceeds as follows. Let Δ denote a set of given axioms (from logical point of view, a conjunction), and let Φ be a formula to be proved. Thus, one is to prove that

$$\Delta \Rightarrow \Phi \tag{1}$$

is a valid formula. In classical resolution method, instead of proving (1), one takes the negation of it, i.e.

$$\Delta \wedge \neg \Phi$$
 (2)

and tries to show that (2) is unsatisfiable. In practice (2) is transformed into so-called clausal form, i.e. to a set (conjunction) of clauses (disjunctions of literals). Now, in order to show that a set of clauses is unsatisfiable, one attempts to derive from it an empty clause (here: always unsatisfiable). The derivation is carried out with use of resolution rule, which preserves logical consequence. Thus, any newly derived clause is a logical consequence of its parent clauses. If an empty clause is eventually derived, the unsatisfiability of the initial set of clauses is proved.

A great advantage of the classical resolution method consists in leaving the set of axioms Δ almost unchanged. In most of practical cases there is a set (conjunction) of separate axioms, and each of them can be converted into clausal form independently from the others. This approach saves computational effort and fits most of classical problems in a naturally efficient way. However, from theoretical point of view, one can also try to prove validity of (1) directly. Below, we present the basic idea in some details.

First, instead of converting a formula to its clausal form one can transform the initial formula to a dual form, consisting of an alternative of conjunctions of literals. Now we shall need the two following definitions.

Definition 1 Let q_1, q_2, \ldots, q_k be literals. A formula $\phi = q_1 \wedge q_2 \wedge \ldots \wedge q_k$ is called a simple formula. Further, let $\phi_1, \phi_2, \ldots, \phi_n$ be some simple formulae. Any formula $\Phi = \phi_1 \vee \phi_2 \vee \ldots \vee \phi_n$ will be called a normal formula.

Thus, a simple formula is a finite conjunction of literals; simple formulae will be frequently denoted with the letters ϕ and ψ . A normal formula is also called in literature a formula in disjunctive normal form (DNF); normal formulae will be most frequently denoted with the letters Φ and Ψ .

For further considerations we assume that the formula to be proved is a normal formula. Note that all the variables in the formula are (implicitly) existentially quantified (dually to classical resolution method). Thus, the formula can be written as

$$\Psi = \psi_1 \vee \psi_2 \vee \psi_3 \vee \dots \psi_n. \tag{3}$$

Let ψ^1 and ψ^2 be two simple formulae of Ψ . Bd-resolution is defined as follows.

Definition 2 (Backward dual resolvent) Let $\psi^1 = \psi^{<1>} \wedge q^1$ and $\psi^2 = \psi^{<2>} \wedge \neg q^2$ and let there exist a most general unifier (mgu) σ for q^1 and q^2 . Then, a formula ψ

$$\psi = \psi^{<1>} \sigma \wedge \psi^{<2>} \sigma \tag{4}$$

will be called a backward dual resolvent (bd-resolvent) of ψ^1 and ψ^2 .

Now, we shall prove that in fact for any bd-resolvent (4), the alternative of the parent formulae ψ^1 and ψ^2 is its logical consequence.

Theorem 1 (Soundness of bd-resolution) Let ψ be the bd-resolvent defined as above (4). Then $\psi^1 \vee \psi^2$ is the logical consequence of ψ .

Proof Let us assume that ψ is satisfied under some interpretation I, i.e. $\models_I \psi$. Thus both $\models_I \psi^{<1>}\sigma$ and $\models_I \psi^{<2>}\sigma$. Obviously, for any pair of complementary literals $q^1\sigma$ and $\neg q^2\sigma$, there is either $\models_I \psi^{<1>}\sigma \wedge q^1\sigma$ or $\models_I \psi^{<2>}\sigma \wedge \neg q^2\sigma$. Thus the alternative of the two above is satisfied. Since changing the terms substituted by σ into existentially quantified variables does not violate satisfiability (existential quantifier introduction), the alternative $\psi^1 \vee \psi^2$ is satisfied under I, Q.E.D.

In order to pass to theorem proving with bd-resolution let us define an idea of derivation with use of bd-resolution. Below, by a factor of a formula we shall understand the result of applying some substitution to the formula in order to replace certain variables with terms. Any formula is logical consequence of any of its factors.

Definition 3 (BD-derivation) A bd-derivation (or derivation, for short) of a simple formula ψ from a normal formula Ψ given by (3) is any sequence of simple formulae $\psi^1, \psi^2, \ldots, \psi^k$, satisfying the following conditions:

- for any i ∈ {1,2,...,k}, ψⁱ is a bd-resolvent or a
 factor of simple formulae either belonging to Ψ or
 generated earlier bd-resolvents,
- $\psi = \psi^k$.

Formula ψ is said to be derived from Ψ .

For convenience, if k = 1 ($\psi \in \Psi$) we shall also say that ψ is derived from Ψ .

Now, we can state the following theorem.

Theorem 2 (Soundness of bd-derivation) Let Ψ be a normal formula defined by (3). If an empty tautology formula (always true, denoted with \top) can be derived from Ψ , then Ψ is valid (is a tautology).

Proof For any simple formula ψ derived from Ψ there is $\psi \models \Psi$. If the empty tautology formula is derived from Ψ , then we have $\top \models \Psi$. Thus Ψ is valid, Q.E.D.

We are not investigating here the problem of completeness of bd-resolution; however, one can expect that due to duality to resolution method the bd-resolution method is potentially complete, as well. The proof can be found in [Ligeza, 1992b].

3 Completeness and generalized resolution

For intuition, a knowledge-based system of rules is *complete* if it "covers" all the possible cases to which the rules are aimed to be applied. Obviously, a set of rules should be complete in order to assure the possibility of dealing with any occurring case. We shall refer to any such case as a *state*. A set of states will be referred to as *situation*. To any state there is assigned a unique formula describing this state, to be called *state formula*. Any situation can have more than one describing formula, to be called *situation formulae*. Of course, if a state s having state formula ϕ belongs to a situation S having situation formula Φ , there is $\phi \models \Phi$.

A definition of completeness requires some more precise statement with regard to logical notions introduced. Generally speaking, from logical point of view, a complete system of rules is one having the potential power of dealing with any occurring state formula. This, in fact, means that the alternative of the rule precondition formulae should constitute an empty tautology formula (always true). In such a case, for any potentially possible formula describing some state s described with state formula ϕ , some of the precondition formulae must "cover" ϕ ; logically, the condition can be written as $\phi \models \Psi_i$, where Ψ_i is a normal formula defining preconditions of some i-th rule. Note however, that the strictly logical completeness will in most cases be too strong for practical applications. This is so, because most of the systems are incomplete with regard to possible state representation [Ligeza, 1992b], and thus, there is no need to "cover" all the "potential" states but only the really existing ones.

With regard to the above considerations the following definition of completeness is stated.

Definition 4 (Completeness) Let us consider a knowledge-based system consisting of a set T of k production rules, each of them having its precondition formula Ψ_i , i = 1, 2, ..., k. The system is:

 logically complete, if and only if the alternative of the preconditions formulae satisfies the following condition:

$$\models \Psi_1 \vee \Psi_2 \vee \ldots \vee \Psi_k$$

²Formal definitions and a detailed presentation of problem concerning states and situations can be found in [Ligeza, 1992b].

(in other words, $\Psi_1 \vee \Psi_2 \vee \ldots \vee \Psi_k$ is a tautology),

• physically complete, if and only if for any existing state s described with formula φ there exists at least one rule in **T** applicable to s, i.e. there is

$$\phi \models \Psi$$

for some $i \in \{1, 2, \ldots, k\}$

Definition 5 (Specific completeness) Consider a knowledge-based system as above. The system is:

specifically logically complete with regard to formula
 \(\bullet\), if and only if the following condition holds

$$\Psi \models \Psi_1 \vee \Psi_2 \vee \ldots \vee \Psi_k$$

specifically physically complete with regard to situation S described with formula Ψ, if and only if for any state s described with formula φ and such that s∈ S (φ ⊨ Ψ) there exists at least one rule in T applicable to s, i.e. the following condition holds for some i

$$\phi \models \Psi_i$$

With regard to the above definitions, a set of rules can be complete in a strictly logical sense (logical completeness) or complete in reference to the particular universe of discourse. Of course, the logical completeness constitutes a stronger requirement; it always implies physical completeness, while the reverse statement is not true.

Both logical and physical completeness can refer to a general case (the former definition) or to a specific case (the latter one). The latter definition provides in fact a weak requirement for completeness. Intuitively speaking, specific completeness means that the scope of possible situations in which the systems is intended to operate is a priori restricted to the one described with formula Ψ . Of course logical (physical) completeness always implies specific logical (physical) completeness. Moreover, specific logical completeness with regard to formula Ψ implies the appropriate specific physical completeness with regard to the situation described with Ψ .

The above definitions of completeness (either logical or physical) are, to certain degree similar; they become the same for S being the situation of all the existing states (for example, for the appropriate situation formula being $\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n$, where $\phi_1, \phi_2, \ldots, \phi_n$ are the state formulae of all the states, or respectively Ψ being the empty tautology formula (always true).

The aim of building complete systems is that it should be possible for the control system to react in and further transform any physical state. This means that the set of rules (the preconditions of them) should cover the whole state-space of the system. Since in practical applications not all the possible to describe states are feasible, it is useful to define a weak completeness, i.e. completeness with regard to some specified formula Ψ (situation S). Of course, situation S should contain all the physically possible states, i.e. Ψ should be chosen in such a way that $\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n \models \Psi$. Simultaneously, the "more detailed" Ψ is, the easier the design of a complete system should be. If all the state formulae are given explicitly, it would be enough to take them as preconditions of the

transformation rules directly (at least potentially; in realistic case the formulae, would be probably very long and there would be quite a large number of them).

Note that the presented above definition of logical completeness does not refer to any specific interpretation. Thus one may regard it as too abstract; in fact, it will be usually too strong - and therefore rarely (or never) satisfied - for realistic systems. In the context of practical applications the definition of physical completeness is much more in place. However, this definition refers implicitly to the intended interpretation, the one under which the formulae describing states refer to the universe of interest (can be read) in a reasonable way.

Let us notice that in the case of verifying physical completeness purely logical methods are likely to be insufficient. These methods are based on syntactic structure of formulae and do not refer to interpretation. Thus, a formula like switch(on)Vswitch(off) will never be regarded as tautology, while with reference to the obvious intended interpretation it is always true - the switch must be either in on or off position. Further, the approaches to completeness verification given in literature [Cragun and Steudel, 1987; Suwa e.t al, 1984] based on inspection of possible cases (states) do, in principle, make use of the physical completeness. Thus, it may be in order to modify the definition of bd-resolution towards covering this case, as well. Below, a formal definition of generalized bd-resolution is proposed.

Definition 6 (Generalized bd-resolution)

Let $\psi^1, \psi^2, \dots, \psi^k$ be some simple formulae, such that $\psi^1 = \psi^{<1>} \wedge q^1, \ \psi^2 = \psi^{<2>} \wedge q^2, \dots, \ \psi^k = \psi^{<k>} \wedge q^k$ and let there exist a most general unifier (mgu) σ for q^1 , q^2, \dots, q^k . Let I denote the intended interpretation. If $\models_I q^1 \sigma \vee q^2 \sigma \vee \dots \vee q^k \sigma$ then, a formula ψ

$$\psi = \psi^{<1>} \sigma \wedge \psi^{<2>} \sigma \wedge \dots \wedge \psi^{} \sigma \tag{5}$$

will be called a generalized backward dual resolvent (generalized bd-resolvent) of ψ^1 , ψ^2 , ..., ψ^k .

We shall prove that with regard to the intended interpretation I the alternative of the parent formulae ψ^1 , ψ^2 , ..., ψ^k is a logical consequence of the generalized bd-resolvent defined by (5).

Theorem 3 (Correctness of generalized bd-resolution) Let ψ be a generalized bd-resolvent defined as above (5). Then, if $\models_I \psi$ then $\models_I \psi^1 \vee \psi^2 \vee ... \vee \psi^k$.

Proof Let us assume that ψ is satisfied under interpretation I, which is the intended one, i.e. $\models_I \psi$. Thus $\models_I \psi^{<1>}\sigma$, $\models_I \psi^{<2>}\sigma$, ..., $\models_I \psi^{<k>}\sigma$. From assumption, $\models_I q^1\sigma \vee q^2\sigma \vee ... \vee q^k\sigma$, and thus for at least one $i \in \{1, 2, ..., k\}$ there is $\models_I q^i\sigma$. Hence, obviously $\models_I \psi^{<i>}\sigma \wedge q^i\sigma$ for at least one $i \in \{1, 2, ..., k\}$, as well. Since turning constants (terms) into existentially quantified variables only "improves" satisfiability (existential quantifier introduction), there is also $\models_I \psi^i$ for at least one $i \in \{1, 2, ..., k\}$ and obviously $\models_I \psi^1 \vee \psi^2 \vee ... \vee \psi^k$, i.e. the alternative of initial formulae is satisfied, Q.E.D.

With regard to the presented logical approach, the following algorithm can be suggested for checking specific completeness with regard to a given formula Ψ . Note that for checking logical completeness it is enough to replace Ψ with \top .

Algorithm for checking completeness

- Build a formula describing the sum of all the appropriate precondition formulae (the normal formula of the form Ψ₁ ∨ Ψ₂ ∨ ... ∨ Ψ_k).
- 2. Perform any possible derivation of bd-resolvent, e.g. by applying the level-saturation method [Chang and Lee, 1973], or any other complete strategy so as to prove that $\Psi \models \Psi_1 \vee \Psi_2 \vee \ldots \vee \Psi_k$.
- 3. If the above logical satisfaction is eventually proved, then the system is complete with regard to Ψ .
- 4. In other case (due to the completeness of the bd-resolution) the system is incomplete. The derived formulae describe the situations which are "served" (i.e. the system is complete with regard to at least the obtained formulae).

Note, that the generation of bd-resolvents is not guaranteed to stop. However, for finite physical systems the number of possible states is finite, and thus the number of situations is finite as well. Since our goal is to "cover" the state space, the situation formulae less than or equally general to some previously generated ones can be eliminated from further derivation. Taking into account that any bd-resolvent is less than or equally general to the alternative of the two "mother" formulae one can expect to eventually 'cut' all the branches of the tree of potential derivations.

4 Examples

Let us analyze an example given in [Cragun and Steudel, 1987], p.635. in TABLE 1. There are given seven rules with the following Boolean preconditions R1-R7.

 $Rl = chatter \land edge_deformation,$

R2= chatter \land redge_deformation \land tool_breakage, R3= chatter \land redge_deformation \land rool_breakage, R4= rchatter \land edge_deformation \land tool_breakage, R5= rchatter \land edge_deformation \land rool_breakage, R6= rchatter \land redge_deformation \land tool_breakage, R7=

The following example inferences can be performed (in parentheses parent formulae are given):

1. chatter A tool_breakage	(R1, R2)
2. chatter ∧ ¬tool_breakage	(R1, R3)
3. edge_deformation ∧ tool_breakage	(R1, R4)
4. edge_deformation ∧ ¬tool_breakage	(R1, R5)
5. chatter	(1, 2)
6. edge_deformation	(3,4)
7. ¬edge_deformation ∧ ¬tool_breakage	(5, R7)
8. ¬edge_deformation A tool_breakage	(5, R6)
9. ¬tool_breakage	(6, 7)
10. tool_breakage	(6, 8)
11. T	(9, 10)

Since an empty tautology formula was deduced with use of bd-resolution, the set of rules is complete.

Now, let us consider a more complex example. Let there be given a rule-based system for automatic control of some object. The rules define the actions which are to be performed while certain input signals occur. There are three input signals, namely input1, input2, and input3. Each input signal can be of low, medium, or high intensity. To denote the level of certain signal we shall use the predicate level(<signal>,<intensity>). There are four control rules which define admissible actions to be performed. The preconditions of the rules are given by the following formulae:

- P1 level(input1,low) \(\Lambda \) \(\tau \) level(input2,high) \(\Lambda \) \(\tau \) level(input3,high),
- P2 level(input1, high) \(\tau\) = level(input2, low) \(\tau\) = \(\tau\) = \(\tau\) = \(\tau\).
- P3 level(input1, medium)
- P4 level(X, low) \land level(Y, high).

We attempt to check if the specified rules are complete, i.e. if in any state, there is at least one rule which can be applied. The bd-derivation can proceed as follows:

- level(input1, low) ∧ ¬level(input2, high); (P1, P4)
 the unifying substitution is {Y/input3}. Factorization is performed by applying substitution {X/input1},
- level(input1,low); (1, P4)
 the unifying substitution is {Y/input2}. Again,
 factorization is performed by applying substitution
 {X/input1},
- level(input1, high) ∧ ¬level(input2, low); (P2, P4)
 the unifying substitution is {X/input3}. Factorization is performed by applying substitution {Y/input1}.
- 4. level(input1, high) (3, P4) the unifying substitution is {X/input2}. Again, factorization is performed by applying substitution {Y/input1}.

At this stage, one can notice that any physical state will be served by the system, since formulae P3, 2, and 4 cover all the possible states (the whole state-space). In fact, the alternative of these formulae is level(input1,low) V level(input1,medium) V level(input1,high), and there are no other possibilities in our system. However, in order to prove this formally, one is to replace P3 with the following formula:

P3' ¬level(input1, low) ∧ ¬level(input1, high),

which, according to the interpretation of the considered system is logically equivalent to P3. Now, the deduction can be completed:

Thus, the above system of rules is complete; i.e. in any state at least one of the rules is applicable.

Note that with use of generalized bd-resolution one can directly generate 6 from P3, 2 and 4. In fact, by putting $q^1 = level(input1, low)$, $q^2 = level(input1, medium)$, and $q^3 = level(input1, high)$ we have $\models_I q^1 \lor q^2 \lor q^3$, where / is the assumed interpretation. The direct application of generalized bd-resolution simplifies the proof and makes it shorter.

In case of incomplete systems, bd-resolution can be used for generating formulae which describe the states for which there are some actions specified, i.e. for proving specific completeness. Some other application examples can be found in [Ligeza, 1992b].

5 Conclusions

The proposed approach based on first order logic and theorem proving dual to resolution method provides theoretical bases for improving the design of knowledge based systems. It suggests a possibility of checking the system for completeness based on first-order logic rather than simple exhaustive inspection of possible parameters/conditions combinations. However, as shown in the above examples, it is likely that for practical applications the proposed algorithms should be equipped with domain-specific knowledge concerning the intended interpretation. Moreover, hierarchization mechanisms and possibly - heuristics would further improve the efficiency of this method.

The advantage of bd-resolution over classical one with respect to completeness verification consists in direct application of the method to formulae describing situations and preconditions in a simple, consistent with intuition way. Moreover, in case of incomplete systems, generation of formulae describing specific completeness (the states actually served) is straightforward.

The approach can be applied to rule-based systems, where precondition formulae are in the form of normal formulae defined here. Logical explicit negation should be used rather than the implicit one. For practical reasons, the proposed generalized bd-resolution taking the advantage of knowing the intended interpretation is proposed; its application can improve efficiency of the proposed paradigm.

The outlined method is primarily aimed at application to "single-layer" rules (such as the ones applied for control in knowledge-based control systems - see, for example [Tzafestas and Ligcjza, 1989, Lige.za, 1992b]); however, it seems that it can be useful for other systems with chaining rules if a hierarchization with regard to a context similar to the one of [Cragun and Steudel, 1987] is performed. Further, it is likely that the combination of the described approach with methods based on exhaustive test can be promising.

References

- [Andert, 1992] Ed P. Andert. Integrated knowledge-based system design and validation for solving problems in uncertain environments. International Journal of Man-Machine Studies, 36:357-373, 1992.
- [Chang and Lee, 1973] Chin-Liang Chang and Richard Char-Tung Lee. Symbolic Logic and Mechanical Theo-

- rem Proving. Academic Press, New York and London, 1973.
- [Cragun and Steudel, 1987] Brian J. Cragun and Harold J. Steudel. A decision-table-based processor for checking completeness and consistency in rule-based expert systems. International Journal of Man-Machine Studies, 26:633-648, 1987.
- [Genesereth and Nilsson, 1987] Michael R. Genesereth and Nils J. Nilsson. Logical Foundations of Artificial Intelligence. M.Kaufmann Publ. Inc., Los Altos, California. 1987.
- [Kuehner, 1971] D.G.Kuehner. A note on the relation between resolution and Maslov's inverse method. In B.Meltzer and D.Michie editors Machine Intelligence 6. Edinburgh University Press, pages 73-76, 1971.
- [Ligqza, 1992a] Antoni Ligeza. A logical formalism for dynamic systems. In R. Trappl editor Cybernetics and Systems Research 92. Proceedings of the Eleventh European Meeting on Cybernetics and Systems Research, Vienna. World Scientific, Singapore, pages 1545-1552, 1992.
- [Ligeza, 1992b] Antoni Ligeza. Logical foundations for knowledge-based control systems. Knowledge representation, reasoning and theoretical properties. Report of Laboratory of Automatics and Systems Analysis (LAAS) Toulouse, France, No. 92269, 1992.
- [Maslov, 1964] S.Ju. Maslov. An inverse method of establishing deducibility in classical predicate calculus (in Russian). Doklady Akademii Nauk SSSR, 159: 17-20, 1964.
- [Maslov, 1968] S.Ju. Maslov. The inverse method of establishing deducibility for logical calculi (in Russian). Trudy Matiematiceskovo Instituta AN SSSR im. W.A.Stteklova, Izdatielswo Nauka, Leningrad, 98:26-87, 1968.
- [Maslov, 1971] S.Ju. Maslov. Proof search strategies for methods of the resolution type. In B.Meltzer and D.Michie editors Machine Intelligence 6. Edinburgh University Press, pages 77-90, 1971.
- [Robinson, 1965] J.A. Robinson. A machine oriented logic based on the resolution principle. Journal of the Association for Computing Machinery, 12(1):23-41, 1965.
- [Suwa et al, 1984] Motoi Suwa et al. Completeness and consistency in a rule-based system. In B.G. Buchanan and E.H. Shortliffe editors Rule-Based Expert Systems, Chapter 8, 159-170. Addison-Wesley, London, 1984.
- [Tepandi, 1990] J. Tepandi. Verification, testing, and validation of rule-based expert systems. Preprints of the 11-th World Congress of International Federation of Automatic Control, Wiinius, pages 162-167, 1990.
- [Tzafestas and Ligeza, 1989] Spyros Tzafestas and Antoni Lige,za. A framework for knowledge based control. Journal of Intelligent and Robotic Systems, 1(4): 407-425.