Lemma Handling in Default Logic Theorem Provers

Thomas Linke[1] and Torsten Schaub[2*]

[1] AG Knowledge-Based Systems, Faculty of Technology, University of Bielefeld, D-33501 Bielefeld, tlinke@techfak.uni-bielefeld.de

[2] Theoretische Informatik, TH Darmstadt, Alexanderstraße 10, D-64283 Darmstadt, schaub@iti.informatik.th-darmstadt.de

Abstract. We develop an approach for lemma handling in automated theorem provers for query-answering in default logics. This work builds on the concept of so-called *lemma default rules*. We show how different forms of such lemmas can be incorporated for reducing computational efforts.

1 Introduction

In automated theorem proving, one often caches *lemmas*, ie. auxiliary propositions used in the demonstration of several propositions, in order to reduce computational efforts. This renders the integration of lemma handling of great practical relevance in automated theorem proving. Unfortunately, such a technique is not applicable in *default logics* [8], since the addition of derived propositions to default theories may change the entire set of conclusions [7].

The central concepts in default logic are *default rules* along with their induced *extensions* of an initial set of facts. Default logic augments classical logic by default rules that differ from standard inference rules in sanctioning inferences that rely upon given as well as absent information. Hence, a default rule $\frac{\alpha : \beta}{\gamma}$ has two types of antecedents: A *prerequisite* α which is established if α is derivable and a *justification* β which is established if β is consistent in a certain way. If both conditions hold, the *consequent* γ is concluded by default. A set of such conclusions (sanctioned by a given set of default rules and by means of classical logic) is called an *extension* of an initial set of facts. At this point it should be clear that the need to incorporate lemmas is even greater in default theorem proving than in standard theorem proving, since computation in default logics not only involves deduction but also consistency checks.

We further develop an approach to lemma handling in default logics, introduced in [9]. The idea is to change the status of a default conclusion whenever it is added to a world-description by turning it into a new default rule. This default rule comprises information about the default proof of the original conclusion and so tells us when its proof is valid or not. In what follows, we elaborate the implementation of this method. This is accomplished by extending an existing algorithm for query-answering in default logics [10]. The whole approach is developed for a variant of default logic, known as *constrained default logic* [3].

2 Lemma Handling in Default Logics

Knowledge is represented in default logics by *default theories* (D, W) consisting of a consistent[3] set of formulas W and a set of default rules D. A *normal default*

[*] On leave from IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

[3] The restriction to consistent set of facts is not really necessary, but it simplifies matters.

theory is restricted to *normal default rules* whose justification is equivalent to the consequent. In any default logic, default rules induce one or more extensions of an initial set of facts.

Since [8], it is well-known that query-answering in default logics is only feasible in the presence of the property of *semi-monotonicity*: If $D' \subseteq D$ for two sets of default rules, then if E' is an extension of (D', W) then there is an extension E of (D, W) such that $E' \subseteq E$. Given this property, it suffices to consider a relevant subset of default rules while answering a query, since applying other default rules would only enlarge or preserve the partial extension at hand. Also, semi-monotonicity implies that extensions are constructible in a truly iterative way by applying one applicable default rule after another. Due to the semi-monotonicity of constrained default logic, one thus obtains the following specification [10]:

Theorem 1. *For a default theory (D, W) and sets of formulas E and C, (E,C) is a constrained extension of (D, W) iff there is some maximal $D' \subseteq D$ that has an enumeration $\langle \delta_i \rangle_{i \in I}$ such that for $i \in I$ the following conditions hold.*

1. $E = \mathit{Th}(W \cup \mathit{Cons}(D'))$ *and* $C = \mathit{Th}(W \cup \mathit{Just}(D') \cup \mathit{Cons}(D'))$
2. $W \cup \mathit{Cons}(\{\delta_0, \ldots, \delta_{i-1}\}) \vdash \mathit{Pre}(\delta_i)$
3. $W \cup \mathit{Cons}(\{\delta_0, \ldots, \delta_{i-1}\}) \cup \mathit{Just}(\{\delta_0, \ldots, \delta_{i-1}\}) \nvdash \neg \mathit{Just}(\delta_i) \vee \neg \mathit{Cons}(\delta_i)$

Condition *(2)* spells out that D' has to be grounded in W. In general, a set of default rules D is *grounded* in a set of facts W iff there exists an enumeration $\langle \delta_i \rangle_{i \in I}$ of D that satisfies Condition *(2)*. Condition *(3)* expresses the notion of *incremental consistency*. Here, the "consistent" application of a default rule is checked at each step, whereas this is done wrt to the final set of constraints in the usual definition of a constrained extension (cf. [3]). A *default proof* D_φ for a formula φ from a default theory (D, W) is a a sequence of default rules $\langle \delta_i \rangle_{i \in I}$ such that $W \cup \{\mathit{Cons}(\delta_i) \mid i \in I\} \vdash \varphi$ and Condition *(2)* and *(3)* in Theorem 1 are satisfied for all $i \in I$. Then, one can show that $\varphi \in E'$ for some constrained extension (E', C') of (D, W) iff φ has a finite default proof D_φ from (D, W).

[9] introduced an approach to lemma handling in Reiter's and constrained default logic. In what follows, we focus on the latter and introduce the notion of a *lemma default rule*: For lemmatizing[4] a default conclusion, we take this conclusion along with one of its default proofs and construct the corresponding lemma default rule in the following way [9].

Definition 2. *Let D_φ be a default proof of a formula φ from default theory (D, W). We define a lemma default rule δ_φ for φ wrt (D, W) as*

$$\delta_\varphi = \frac{: \bigwedge_{\delta \in D_\varphi} \mathit{Just}(\delta) \wedge \bigwedge_{\delta \in D_\varphi} \mathit{Cons}(\delta)}{\varphi}.$$

[9] shows that the addition of lemma default rules does not alter the constrained extensions of a given default theory: Let δ_φ be a lemma default rule for φ. Then, (E,C) is a constrained extension of (D, W) iff (E,C) is a constrained extension of $(D \cup \{\delta_\varphi\}, W)$. So, the approach provides a simple solution for generating and using lemma defaults. Whenever we lemmatize a conclusion, we change its representation into a default rule and add it to the default rules of a considered

[4] Ie. to introduce a derivable theorem as a lemma by adding it to the initial theory.

default theory. So lemma defaults are abbreviations for the default inferences needed for deriving a conclusion. This approach is discussed in detail in [9]. There, it is also contrasted with the one taken in [2].

In fact, we do not have to transform a default conclusion into a lemma default rule if we focus on constrained extensions that are "compatible" with the underlying constraints imposed by the conclusion's default proof:

Theorem 3. *Let δ_φ be a lemma default rule for φ wrt a default theory (D, W). Then, we have for all sets of formulas E and C where $C \cup \mathit{Just}(\delta_\varphi) \cup \{\varphi\}$ is consistent that (E,C) is a constrained extension of (D, W) iff (E,C) is a constrained extension of $(D, W \cup \{\varphi\})$.*

This result justifies the use of so-called *dynamic lemmas*, which are applicable in the course of a proof search without consistency checks due to the (previously established) consistency of $\mathit{Just}(\delta_\varphi) \cup \{\varphi\}$ with the default proof segment at hand. In contrast to dynamic lemmas, we call a lemma default rule δ_φ *static*, if it was generated in an independent default proof. In such a case, merely the consistency with the default proof segment at hand has to be checked in order to apply δ_φ. Algorithmically, let $\mathcal{A}(\varphi, D, W)$ be a boolean algorithm returning true iff φ is in some constrained extension (E,C) of (D, W). Then, a lemma default rule δ_φ of (D, W) is supplied as a *static* lemma to $\mathcal{A}$ if it is used as a default rule, as in $\mathcal{A}(\varphi, D \cup \{\delta_\varphi\}, W)$. If additionally for some $D' \subseteq D$, we have $W \cup \mathit{Just}(D') \cup \mathit{Cons}(D') \cup \mathit{Just}(\delta_\varphi) \cup \{\varphi\}$ is consistent, then δ_φ is *dynamically* usable, via a "call'' $\mathcal{A}(\varphi, D, W \cup \{\varphi\})$. With Theorem 3 we have $\mathcal{A}(\varphi, D \cup \{\delta_\varphi\}, W)$ iff $\mathcal{A}(\varphi, D, W \cup \{\varphi\})$ for dynamic lemma default rules.

Consider for example the statements "profs are typically adults", "adults are typically employed", "employed people typically have money", and "employed people having money typically buy cars", which yields the following theory.

$$\left(\left\{\tfrac{P:A}{A}, \tfrac{A:E}{E}, \tfrac{E:M}{M}, \tfrac{E\wedge M:B}{B}\right\}, \{P\}\right) \qquad (1)$$

Consider the query B, asking whether profs buy cars. Proving this, by definition of a default proof, is to form a sequence $\langle\delta_i\rangle_{i\in I}$ of default rules such that B follows from $\{P\} \cup \{\mathit{Cons}(\delta_i) \mid i \in I\}$. For proceeding in a query-oriented manner, we have to form this sequence by starting with the rightmost default rule in sequence $\langle\delta_i\rangle_{i\in I}$ and working our way "down" to the facts while satisfying Condition *(2)* and *(3)* of Theorem 1. So let us put the default rule $\frac{E\wedge M:B}{B}$ at the end of the sequence. This can be done, if we can derive $(E \wedge M)$ from the remaining default rules together with $\{P\}$. That is, we have to prove in turn E and, independently, M. This can be done by means of the default proofs $D_E = \left\langle\frac{P:A}{A}, \frac{A:E}{E}\right\rangle$ and $D_M = \left\langle\frac{P:A}{A}, \frac{A:E}{E}, \frac{E:M}{M}\right\rangle$ for E and M, respectively. For the query B, we then obtain the default proof $D_B = \left\langle\frac{P:A}{A}, \frac{A:E}{E}, \frac{E:M}{M}, \frac{E\wedge M:B}{B}\right\rangle$. Observe that this approach bears a certain redundancy due to $D_E \subseteq D_M$. That is, if we build the default proof D_B in the aforementioned way, we consider the subsequence $\langle\frac{P:A}{A}, \frac{A:E}{E}\rangle$ twice. Consequently, we also have to check conditions *(2)* and *(3)* of Theorem 1 twice for each default rule in this default proof segment. However, this redundancy can be avoided by means of lemma default rules. Suppose, we first build the default proof $D_E = \left\langle\frac{P:A}{A}, \frac{A:E}{E}\right\rangle$ and lemmatize the conclusion E afterwards. This yields the lemma default rule $\delta_E = \frac{:A\wedge E}{E}$. Notably, this lemma default rule can be used for simplifying the second default proof

$D_M = \left\langle \frac{P:A}{A}, \frac{A:E}{E}, \frac{E:M}{M} \right\rangle$ to $D'_M = \left\langle \frac{:A\wedge E}{E}, \frac{E:M}{M} \right\rangle$

without any consistency checks. That is, not even the justification of δ_E, $A \wedge E$, has to be checked for consistency, since the underlying default proof D_E constitutes a viable segment of the actual default proof. In fact, the treatment of $\frac{:A\wedge E}{E}$ in the course of the proof search for B gives an example for handling dynamic lemma default rules. That is, a dynamic lemma default rule is applicable without any consistency checks due to the contribution of the underlying default proof to the default proof of the original query. Clearly, this approach is of great practical relevance since it reduces computational efforts in a significant way.

In addition to using δ_E as a dynamic lemma, we can generate and keep lemma default rules, like δ_E and $\delta_M = \frac{:A\wedge E\wedge M}{M}$, for later usage. Then, during a later proof search, we can use δ_E and δ_M as static default rules with the usual consistency check. Notably, in any case, none of the default proofs underlying δ_E and δ_M, namely D_E and D_M, have then to be reconsidered. Note also that the use of static lemma default rules leads to even shorter proofs than in the case of dynamic lemmas, because we can directly jump to the desired conclusions.

3 Lemma Handling while Query-Answering

The general idea of our algorithmic approach is to proceed in a query-oriented manner. For this, we extend the approach taken in [10]. In fact, [10] gives an algorithm following the line of Theorem 1. Even though this algorithm relies on the connection method [1], it can as well be seen as an algorithm based on model elimination [6]. Both deduction methods allow for testing the unsatisfiability of formulas in conjunctive normal form (CNF) and can be outlined as follows.[5] The proof procedures are carried out by means of two distinct inference operations, called *extension* and *reduction* operation. An extension operation amounts to Prolog's use of input resolution. That is, a subgoal, say K, is resolved with an input clause, say $\{J, \neg K, L\}$, if the subgoal is complementary to one of the literals in the selected clause, here $\neg K$. This yields two new subgoals, J and L. The reduction operation renders the inference system complete: If the current subgoal is complementary to one of its ancestor subgoals, then the current subgoal is solved. In the connection method the ancestor goals are accumulated in a so-called *path*, which intuitively corresponds to a path through a CNF obtained by taking the ancestor subgoal from each previous input clause.

In what follows, we refine the approach taken in [10] in order to incorporate the generation and application of lemma default rules. For this purpose, we redefine the predicate $\mathit{compl}(p, C_W, C_D)$ used in [10] as a declarative description of a query-answering algorithm in default logics. But first let us introduce some notions we use later on. We say a default theory (D, W) is in *atomic format* if all formulas occuring in the defaults in D are atomic. A general method to transform default theories in their atomic format is described in [10]. In what follows, we consider only propositional default theories in atomic format.

[5] For a detailed description the reader is referred to [1, 6].

In order to find out whether a formula φ is in some extension of a default theory (D, W) we proceed as follows: First, we transform the default rules in D into their sentential counterparts. This yields a set of indexed implications: $W_D = \{\alpha_\delta \rightarrow \gamma_\delta \mid \frac{\alpha_\delta : \beta_\delta}{\gamma_\delta} \in D\}$. Second, we transform both W and W_D into their clausal forms, C_W and C_D. The clauses in C_D, like $\{\neg\alpha_\delta, \gamma_\delta\}$, are called *$\delta$-clauses*; accordingly, *lemma δ-clauses* are simple unit-clauses, like $\{\gamma_\delta\}$; all other clauses like those in C_W are referred to as *ω-clauses.* For any set of formulas S, let C_S be the set of clauses corresponding to the CNF of S.

Now, let us turn to our extension of the algorithm proposed in [10]: We define a predicate *compl* such that $compl(\neg\varphi, C_W, C_D)$ is true iff there is a default proof of φ from (D, W). The first argument of the predicate is a set of literals describing a partial path containing all ancestor goals, the second argument represents a set of ω-clauses, and the last argument accounts for δ-clauses. For incorporating lemma generation, we define a lemma operator as follows. Let (D, W) be default theory and S some set of literals. For $C_{D'} \subseteq C_D$, we define

$$\mathcal{C}_W^S(C_{D'}) = \{\{\varphi\} \mid \varphi \in S \text{ and } compl(\{\neg\varphi\}, C_W, C_{D'})\}.$$

For distinguishing the components of the default theory, say (D°, W°), from formal parameters, let C_W° and C_D° be fixed sets of ω- and δ-clauses corresponding to the original sets W° and D°, respectively. C_W and C_D function as parameters.

Definition 4. *Let $C_W \subseteq C_W^\circ$ be a set of ω-clauses, $C_D \subseteq C_D^\circ$ be a set of δ-clauses, $C_L \subseteq \mathcal{C}_{W^\circ}^S(C_D^\circ)$ be a set of (static) lemma δ-clauses and S be a set of literals. Let $C_{DL} = C_D \cup C_L$ be the set of all δ-clauses. Then, we define $compl(p, C_W, C_{DL})$ relative to C_W° as follows.*

1. *If $C_W \cup C_{DL} = \emptyset$ then $compl(p, C_W, C_{DL})$ is false.*
2. *If $C_W \neq \emptyset$ and $c \in C_W$ then $compl(p, C_W, C_{DL})$ is true iff the following two conditions hold for $c = c_1 \cup c_2$.*
   - *(a) for all $K \in c_1$, K is complementary to some literal of p.*
   - *(b) for all $K \in c_2$, there is a set of δ-clauses $C_{D(K)} \subseteq C_{DL}$ and a set of (dynamic) lemma δ-clauses $C_L^\star \subseteq \mathcal{C}_{W^\circ}^S(\bigcup_{K\in c_2} C_{D(K)})$ such that the following two conditions hold.*
     - *i. $compl(p \cup \{K\}, (C_W \setminus \{c\}) \cup C_L^\star, C_{D(K)})$ is true.*
     - *ii. $compl(Just(\bigcup_{K\in c_2} D(K)), C_W^\circ \cup \bigcup_{K\in c_2} C_{D(K)}, \emptyset)$ is false.*
3. *If $C_D \neq \emptyset$ and $c \in C_D$ then $compl(p, C_W, C_{DL})$ is true iff the following two conditions hold for $c = \{\neg\alpha_\delta, \gamma_\delta\}$.*
   - *(a) γ_δ is complementary to some literal of p.*
   - *(b) There is a set of δ-clauses $C_{D(\neg\alpha_\delta)} \subseteq C_{DL}$ and a set of (dynamic) lemma δ-clauses $C_L^\star \subseteq \mathcal{C}_{W^\circ}^S(C_{D(\neg\alpha_\delta)})$ such that $\{\neg\alpha_\delta, \gamma_\delta\} \in C_{DL} \setminus C_{D(\neg\alpha_\delta)}$ and the following two conditions hold.*
     - *i. $compl(\{\neg\alpha_\delta\}, C_W^\circ \cup C_L^\star, C_{D(\neg\alpha_\delta)})$ is true.*
     - *ii. $compl(Just(D(\neg\alpha_\delta) \cup \{\delta\}), C_W^\circ \cup C_{D(\neg\alpha_\delta)} \cup \{\{\neg\alpha_\delta, \gamma_\delta\}\}, \emptyset)$ is false.*
4. *If $C_L \neq \emptyset$ and $c \in C_L$ then $compl(p, C_W, C_{DL})$ is true iff the following two conditions hold for $c = \{\gamma_\delta\}$.*
   - *(a) γ_δ is complementary to some literal of p.*
   - *(b) $compl(Just(\delta), C_W^\circ \cup \{\{\gamma_\delta\}\}, \emptyset)$ is false.*

The preceding algorithm is an extension of the standard algorithm for the connection method given in [4]. In fact, the first two conditions provide a sound and complete algorithmic characterization of the standard connection method (see [4] for details) when discarding all δ-clauses. While Condition *(1)* accounts for the limiting case, Condition *(2)* deals with ω-clauses. *(2a)* corresponds to the aforementioned reduction operation, while *(2bi)* is an extension operation. Condition *(2bii)* was added in [10] in order to guarantee the compatibility of multiple subproofs found in *(2bi)*. Condition *(3)* deals with δ-clauses:[6] *(3a)* corresponds to *(2a)* and says that the consequent of a default rule γ_δ can be used for query-answering as any other proposition—provided *(3bi)* and *(3bii)* are satisfied. In fact, *(3bi)* "implements" Statement (*2*) in Theorem 1 and ensures that the prerequisite α_δ of a default rule is derivable in a non-circular way. Correspondingly, *(3bii)* "implements" Statement (*3*) in Theorem 1. This treatment is discussed in detail in [10], so that we focus now on the treatment of lemma default rules.

While static lemma δ-clauses are globally supplied by C_L, dynamic lemma δ-clauses are generated "on the fly" from the default proofs at hand. Dynamic lemmas are represented by $C_L^\star$; they are formed by considering all δ-clauses used in proving the set of subgoals under consideration, namely those in c_2 and $\{\neg\alpha_\delta\}$ in *(2b)* and *(3b)*, respectively. In this way, the resulting lemma δ-clauses can be treated in the subproof as if they were standard ω-clauses. The consistency of the corresponding justifications is ensured by *(2bii)* and *(3bii)*. Accordingly, dynamic lemma δ-clauses are added to the current ω-clauses in *(2bi)* and *(3bi)* via $(C_W \setminus \{c\}) \cup C_L^\star$ and $C_W^\circ \cup C_L^\star$, respectively.

Condition *(4)* deals with the application of static lemma δ-clauses. The treatment is analogous to that of δ-clauses in *(3)* with the exception that no prerequisite has to be proven. Consequently, there are no subproofs to be accomplished, which renders further lemma generation obsolete. In contrast to the treatment of dynamic lemmas in *(3)*, the consistency of the justification of the corresponding lemma default rule has to be checked in *(4b)*.

Let us illustrate $compl(p, C_W, C_D)$ by reconsidering the derivation of B from default theory (1), given at the end of Section 2. Since algorithm 4 is defined for default theories in atomic format, we partially transform our initial default theory in atomic format: For a new atom EM, the default theory

$$\left(\left\{\tfrac{P:A}{A}, \tfrac{A:E}{E}, \tfrac{E:M}{M}, \tfrac{EM:B}{B},\right\}, \{P\} \cup \{E \wedge M \rightarrow EM\}\right) \tag{2}$$

is a conservative extension of default theory (1). This theory yields the set of original ω-clauses $C_W^\circ = \{\{P\}, \{\neg E, \neg M, EM\}\}$, so that we have to show that

$$compl(\{\neg B\}, C_W^\circ, \{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}, \{\neg E_{\delta_3}, M_{\delta_3}\}, \{\neg EM_{\delta_4}, B_{\delta_4}\}\}). \tag{3}$$

We select clauses in a connection-driven way. Thus, we select the δ-clause $\{\neg EM_{\delta_4}, B_{\delta_4}\}$ since B_{δ_4} is complementary to the literal $\neg B$ on the active path. This establishes Condition *(3a)*. Next, we have to verify *(3b)*. For this, we have to find a subset $C_{D(\neg EM_{\delta_4})}$ of the remaining δ-clauses in

[6] Note that default (sub)proofs are given as existentially quantified sets of default rules, such as $C_{D(K)}$ or $C_{D(\neg\alpha_\delta)}$ in *(2b)* and *(3b)*, respectively, in order to retain maximum degrees of freedom.

$\{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}, \{\neg E_{\delta_3}, M_{\delta_3}\}\}$ satisfying *(3bi)* and *(3bii)*. For illustration, we direct our subsequent choices along the line sketched by the derivation in Section 2: For $C_{D(\neg EM_{\delta_4})} = \{\{\neg P_{\delta_1}, A_{\delta_1}\},\ \{\neg A_{\delta_2}, E_{\delta_2}\},\ \{\neg E_{\delta_3}, M_{\delta_3}\}\}$, we then obtain for *(3bi)* and *(3bii)*: [7]

$$\mathit{compl}(\{\neg EM_{\delta_4}\}, C_W^\circ, \{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}, \{\neg E_{\delta_3}, M_{\delta_3}\}\}) \text{ is true and } \quad (4)$$

$$\mathit{compl}(\emptyset, C_W^\circ \cup \{\{\neg P_{\delta_1}, A_{\delta_1}\},\{\neg A_{\delta_2}, E_{\delta_2}\},\{\neg E_{\delta_3}, M_{\delta_3}\},\{\neg EM_{\delta_4}, B_{\delta_4}\}\}, \emptyset) \text{ is false} \quad (5)$$

As can be easily seen, (5) is reducible by successive applications of *(2)* to $\mathit{compl}(\{P, A_{\delta_1}, E_{\delta_2}, M_{\delta_3}, EM, B_{\delta_4}\}, \emptyset, \emptyset)$ is false.

For showing (4), namely *(3bi)*, we select the ω-clause $\{\neg E, \neg M, EM\}$ from C_W° because EM is complementary to $\neg EM_{\delta_4}$ on the active path in (4); this establishes *(2a)*. For showing *(2b)*, we have to determine $C_{D(\neg E)}$, $C_{D(\neg M)}$ and $C_L^\star$ such that *(2bi)* and *(2bii)* hold. Observe that *(2bi)* has to be separately satisfied by $C_{D(\neg E)}$ and $C_{D(\neg M)}$, while *(2bii)* has to be jointly satisfied by $C_{D(\neg E)}$ and $C_{D(\neg M)}$. Since the choice of $C_L^\star$ is an arbitrary one, let us illustrate how $C_L^\star \subseteq \mathcal{C}_{W^\circ}^S(C_{D(\neg E)} \cup C_{D(\neg M)})$ can be successively build along the lines sketched in Section 2.

Let $C_{D(\neg E)} = \{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}\}$ and let $C_L^\star$ be an arbitrary set of lemma δ-clauses at this point. Then, we have to show *(2bi)*:

$$\mathit{compl}(\{\neg EM_{\delta_4}, \neg E\}, \{\{P\}\} \cup C_L^\star, \{\{\neg P_{\delta_1}, A_{\delta_1}\},\ \{\neg A_{\delta_2}, E_{\delta_2}\}\}) \text{ is true.} \quad (6)$$

By continuing along the same argumentation leading from (3) to (4) and (5), we first select the δ-clause $\{\neg A_{\delta_2}, E_{\delta_2}\}$; second we select the δ-clause $\{\neg P_{\delta_1}, A_{\delta_1}\}$ while applying *(3)* twice. In this way, we confirmed (6), viz. *(2bi)*, for $C_{D(\neg E)}$ without any usage of lemma δ-clauses. This corresponds intuitively to proving E from $\{P\}$ by means of the default proof D_E in Section 2.

For choosing $C_{D(\neg M)}$ (and $C_L^\star$), we have two extreme possibilities. First let

$$C_{D(\neg M)} = \{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}, \{\neg E_{\delta_3}, M_{\delta_3}\}\} \quad \text{and} \quad C_L^\star = \emptyset\ . \quad (7)$$

In this case, no dynamic lemmas are used. On the other hand, we have established (6) without any use of dynamic lemmas. This allows us to draw dynamic lemma δ-clauses according to the definition of $\mathcal{C}_W^S$. In this way, we can take advantage of the fact that we have proven E in (6) and obtain the following alternative choice for $C_{D(\neg M)}$ (and $C_L^\star$):

$$C_{D(\neg M)} = \{\{\neg E_{\delta_3}, M_{\delta_3}\}\} \quad \text{and} \quad C_L^\star = \{\{E\}\}. \quad (8)$$

Depending on our choice of $C_{D(\neg M)}$ and $C_L^\star$, we are faced with the following conditions establishing *(2bi)*:

$$\mathit{compl}(\{\neg EM_{\delta_4}, \neg M\}, \{\{P\}\}, \{\{\neg P_{\delta_1}, A_{\delta_1}\},\{\neg A_{\delta_2}, E_{\delta_2}\},\{\neg E_{\delta_3}, M_{\delta_3}\}\}) \text{ is true} \quad (9)$$

$$\mathit{compl}(\{\neg EM_{\delta_4}, \neg M\}, \{\{P\} \cup \{E\}\}, \{\{\neg E_{\delta_3}, M_{\delta_3}\}\}) \text{ is true} \quad (10)$$

The latter can be reduced to $\mathit{compl}(\{\neg EM_{\delta_4}, \neg M, P, E\}, \emptyset, \{\{\neg E_{\delta_3}, M_{\delta_3}\}\})$ by successive applications of *(2)*, while the former yields $\mathit{compl}(\{\neg EM_{\delta_4}, \neg M, P\}, \emptyset, \{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}, \{\neg E_{\delta_3}, M_{\delta_3}\}\})$. In this way, we can solve (10) by means of *one* default inference, namely Condition *(3)*, due to the usage of dynamic lemmas, while we are faced with *three* default inferences in (9) if we fail to notice dynamic lemmas.

[7] Since we deal with normal default rules no justifications have to be added in (5) to the path.

Finally, we have to check *(2bii)* for $C_{D(\neg E)}$ and $C_{D(\neg M)}$: [8]

$compl(\emptyset, C_W^\circ \cup \{\{\neg P_{\delta_1}, A_{\delta_1}\}, \{\neg A_{\delta_2}, E_{\delta_2}\}, \{\neg E_{\delta_3}, M_{\delta_3}\}\}, \emptyset)$ is false.

As above, this can be confirmed by successive applications of *(2)* (no matter which choice, (7) or (8)). That is, $compl(\{P, A_{\delta_1}, E_{\delta_2}, M_{\delta_3}, EM\}, \emptyset, \emptyset)$ is false. Now, we have shown items (4) and (5) confirming Condition *(3bi)* and *(3bii)* so that our proof of B is completed.

Finally, we obtain the following result showing that our incremental algorithm with lemma handling is correct and complete for query-answering in constrained default logic:

Theorem 5. *Let (D, W) be a default theory in atomic format, φ an atomic formula and L a set of lemma default rules. Then, $\varphi \in E$ for some constrained extension (E,C) of (D, W) iff $compl(\{\neg\varphi\}, C_W, C_D \cup C_L)$ is true, where C_W is the matrix of W, C_D is the matrix of W_D and C_L is the matrix of L.*

4 Conclusion

We have implemented our approach and tested it on numerous examples. Due to space restrictions however we had to remove this part of the paper. The experimental results are given in [5]. What has been achieved? One of the original postulates of default formalisms was to "jump to conclusions" in the absence of information. But since the computation of default conclusions involves not only deduction but also expensive consistency checks, the need to incorporate lemmas is even greater in default theorem proving than in standard theorem proving. Hence, default lemmas can be seen as a step in this direction.

References

1. W. Bibel. *Automated Theorem Proving.* Vieweg, 1987.
2. G. Brewka. Cumulative default logic: In defense of nonmonotonic inference rules. *Artificial Intelligence*, 50(2):183–205, 1991.
3. J. Delgrande, T. Schaub, W. Jackson. Alternative approaches to default logic. *Artificial Intelligence*, 70(1–2):167–237, 1994.
4. E. Eder. *Relative Complexities of First Order Calculi.* Vieweg, 1992.
5. Th. Linke and T. Schaub. Lemma handling in default logic theorem provers. Technical report, Faculty of Technology, University of Bielefeld, 1995. (in preparation).
6. D. Loveland. *Automated Theorem Proving: A Logical Basis.* North-Holland, 1978.
7. D. Makinson. General theory of cumulative inference. In M. Reinfrank et al, eds, *Proc. 2nd Int. Workshop on Non–Monotonic Reasoning*, 1–18. Springer, 1989.
8. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.
9. T. Schaub. On constrained default theories. In B. Neumann, ed, *Proc. of the European Conf. on Artificial Intelligence*, p. 304–308. Wiley, 1992.
10. T. Schaub. A new methodology for query-answering in default logics via structure-oriented theorem proving. *Journal of Automated Reasoning*, 1995. Forthcoming.

[8] Since we deal with normal default rules no justification have to be added in *(2bii)* to the path.