

# Uniform interpolation by resolution in modal logic

Andreas Herzig and Jérôme Mengin<sup>1</sup>

**Abstract.** The problem of computing a uniform interpolant of a given formula on a sublanguage is known in Artificial Intelligence as variable forgetting. In propositional logic, there are well known methods for performing variable forgetting. Variable forgetting is more involved in modal logics, because one must forget a variable not in one world, but in several worlds. It has been shown that modal logic  $K$  has the uniform interpolation property, and a method has recently been proposed for forgetting variables in a modal formula (of  $\mu$ -calculus) given in disjunctive normal form. However, there are cases where information comes naturally in a more conjunctive form. In this paper, we propose a method, based on an extension of resolution to modal logics, to perform variable forgetting for formulae in conjunctive normal form, in the modal logic  $K$ .

## 1 Introduction

An interpolant of logical formulae  $\phi$  and  $\psi$  such that  $\phi \models \psi$  in a logic  $L$  is a formula  $\chi$  that contains only variables that appear in both  $\phi$  and  $\psi$ , and such that  $\phi \models \chi$  and  $\chi \models \psi$ . A uniform interpolant of  $\phi$  with respect to a sublanguage  $\mathcal{L}'$  of the language of  $\phi$  is a formula  $\chi \in \mathcal{L}'$  entailed by  $\phi$  that can act as an interpolant for any  $\psi \in \mathcal{L}'$ : if  $\phi \models \psi$ , then  $\chi \models \psi$ . In other words,  $\chi$  behaves like  $\phi$  when  $\mathcal{L}'$  is concerned, in the sense that  $\psi$  has the same  $\mathcal{L}'$ -logical consequences as  $\phi$  [4].

When the language  $\mathcal{L}'$  is defined as the set of formulae that contain no variable of a given set  $P$ , the problem of computing a uniform interpolant of  $\phi$  on  $\mathcal{L}'$  is known in Artificial Intelligence as *variable forgetting*. In propositional logic, there are well known methods for performing variable forgetting [8, 9].

Variable forgetting is more involved in modal logics, because one must forget a variable not in one world, but in several worlds. It has been shown that modal logic  $K$  has the uniform interpolation property [11, 6] (examples of logics that do not have the uniform interpolation property include classical first order logic and S4 [7]).

[4] propose a simple method for forgetting variables in a modal formula (of  $\mu$ -calculus) given in disjunctive normal form. However, there are cases where information comes naturally in a more conjunctive form. In this paper, we propose a method, based on an extension of resolution to modal logics, to perform variable forgetting for formulae in conjunctive normal form, in the modal logic  $K$ .

In the next section, we briefly recall the syntax and semantics of  $K$ , and the definitions of disjunctive and conjunctive normal forms in this logic. In section 3, we recall Enjalbert and Fariñas' resolution system for  $K$ . In section 4, we explain how their resolution system can be used to perform variable forgetting in that logic.

## 2 The logic $K$

The language of  $K$ , over a set of propositional variables  $\mathcal{P}$ , is the smallest set  $\mathcal{L}$  of formulae that contains  $\mathcal{P}$  and is closed under conjunction  $\wedge$ , negation  $\neg$  and necessity  $\Box$ .

A *pointed model* for  $K$  is a tuple  $m = (W, R, I, w)$ , where  $W$  is a non-empty set,  $R$  is a binary relation over  $W$ ,  $I$  assigns to every  $p \in \mathcal{P}$  and every  $w' \in W$  a value  $I(p, w') \in \{\text{true}, \text{false}\}$ .

Satisfaction of formula  $\phi$  by model  $m = (W, R, I, w)$  is defined by induction as follows:

- $m \models p$  if and only if  $I(p, w) = \text{true}$ , for every  $p \in \mathcal{P}$ ;
- $m \models \neg\phi$  if and only if  $m \not\models \phi$ ;
- $m \models \phi \wedge \psi$  if and only if  $m \models \phi$  and  $m \models \psi$ ;
- $m \models \Box\phi$  if and only if  $(W, R, I, w') \models \phi$  for every  $w' \in W$  such that  $wRw'$ .

A set of formulae  $S$  is said to entail formula  $\psi$ , written  $S \models \psi$ , if for every pointed model  $m$  that satisfies every formula in  $S$ ,  $m$  satisfies  $\psi$  too. In this case, we will also say that  $\psi$  is a logical consequence of  $S$ . We will write  $S \not\models \psi$  when this is not the case. Given a finite set of formulae  $\{\phi_1, \dots, \phi_n\}$ , we will sometimes write  $\phi_1, \dots, \phi_n \models \psi$  instead of  $\{\phi_1, \dots, \phi_n\} \models \psi$ .

As usual, two other connectors  $\vee$  and  $\Diamond$  are introduced and defined as abbreviations:  $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$  and  $\Diamond\phi \equiv \neg\Box\neg\phi$ .

For that local definition of logical consequence, the deduction theorem holds for finite  $S$ :  $S \models \phi$  if and only if  $\{\} \models \neg\bigvee S \vee \phi$ .

There are various ways to define conjunctive and disjunctive normal forms in modal logic (see e.g. [3, 4, 5]). In this paper, we will consider normal forms close to the ones defined by Enjalbert and Fariñas. Literals, clauses and terms can be defined using a grammar in BNF:

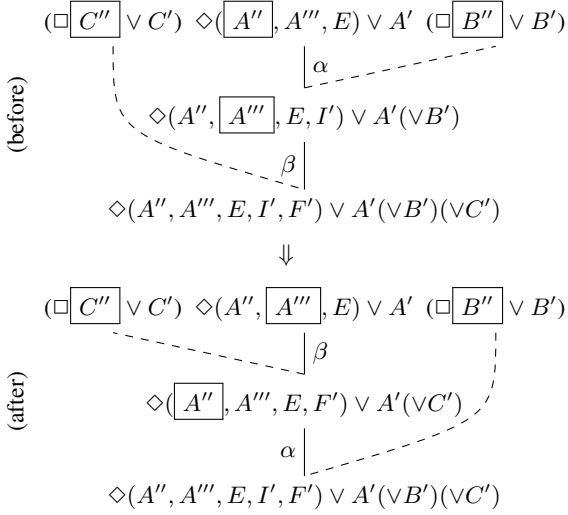
$$\begin{aligned}
 \text{Lit}C & ::= p \mid \neg p \mid \Box\text{Clause} \mid \Diamond\text{CNF} \\
 \text{Lit}T & ::= p \mid \neg p \mid \Box\text{DNF} \mid \Diamond\text{Term} \\
 \text{Clause} & ::= \text{Lit}C \mid \text{Clause} \vee \text{Clause} \mid \perp \\
 \text{Term} & ::= \text{Lit}T \mid \text{Term} \wedge \text{Term} \mid \top \\
 \text{CNF} & ::= \text{Clause} \mid \text{CNF}, \text{CNF} \\
 \text{DNF} & ::= \text{Term} \mid \text{DNF} \vee \text{DNF}
 \end{aligned}$$

According to this definition, clauses are disjunctions of literals of type ‘‘C’’ (for ‘‘Clause’’): such a literal is either a propositional literal, or a clause behind a  $\Box$ , or a set/conjunction of clauses behind a  $\Diamond$ . A CNF is then a set/conjunction of clauses. This corresponds exactly to the definition of clauses used by Enjalbert and Fariñas. A term is the dual of a clause: it is a conjunction of literals of type ‘‘T’’, where such a literal is either a propositional literal, or a term behind a  $\Diamond$ , or a disjunction of terms behind a  $\Box$ . Every formula of  $K$  as an equivalent DNF and an equivalent CNF.

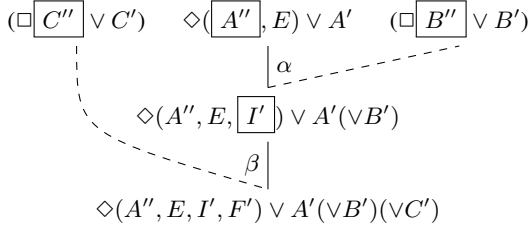
<sup>1</sup> Institut de Recherche en Informatique de Toulouse. e-mail: herzig, mengin@irit.fr



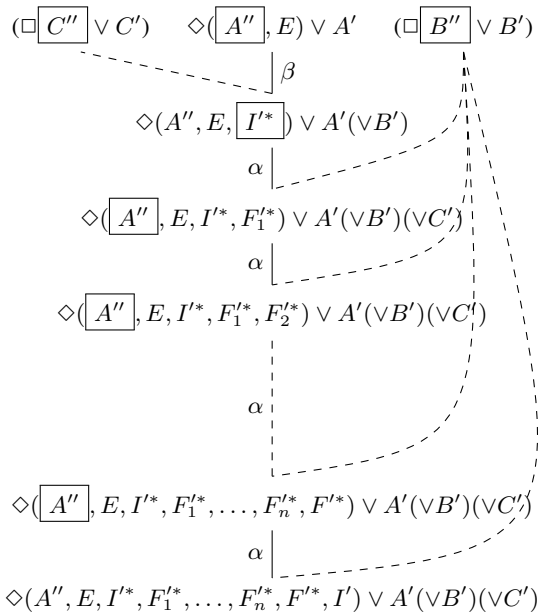




If the  $\beta$ -resolution does involve  $I'$ , then we'll have to use some inductive argument in order to conclude:  $A$  is of the form  $A = \diamond(E, A'') \vee A'$ , and we have  $A''(, B'') \Rightarrow_{\alpha} I'$  and  $I'(, C'') \Rightarrow_{\beta} F'$ :

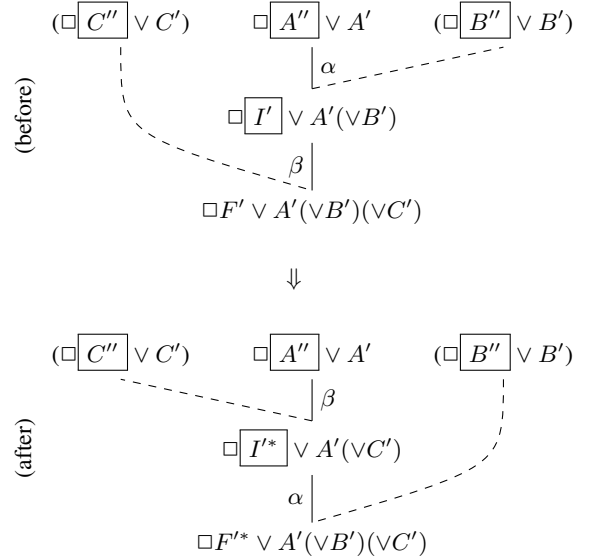


If we have proved that the result of the lemma holds for every resolution whose derivation height is strictly less than that of our current  $\alpha$ -resolution, then we can assume it holds for  $A''(, B'') \Rightarrow_{\alpha} I'$ , in which case there exist  $I^*$ ,  $F_1^*$ , ...,  $F_n^*$  (for some  $n \geq 0$ ) and  $F'^*$  such that  $F'^*$  subsumes  $F'$  and  $A''(, C'') \Rightarrow_{\beta} I^*$  and  $I^*(, B'') \Rightarrow_{\alpha} F_1^*$  and  $F_1^*(, B'') \Rightarrow_{\alpha} F_2^*$  and ... and  $F_n^*(, B'') \Rightarrow_{\alpha} F'^*$ . Now, we can use another resolution to obtain  $I'$ , since we still have  $A''(, B'') \Rightarrow_{\alpha} I'$ . Thus we have:

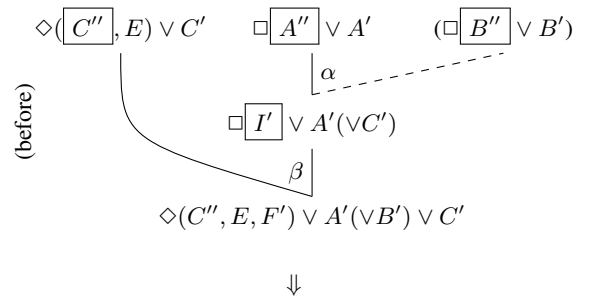


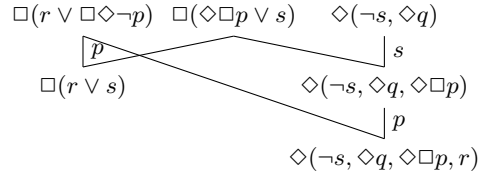
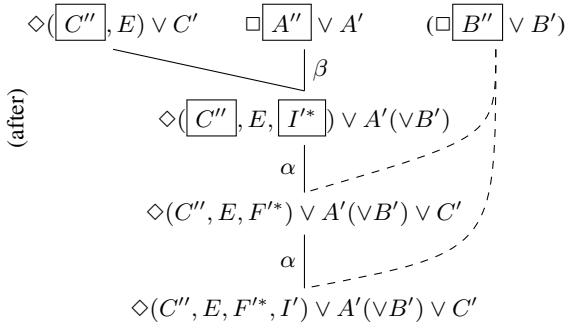
We have used an inductive argument in the preceding paragraph. The base case corresponds to an  $\alpha$ -resolution whose derivation height is zero: these are the propositional resolutions and the  $\square\perp$  resolution, we have already proved that the result holds for these cases.

Let us turn now to the case where there is no  $\diamond$ -literal involved in the  $\alpha$ -resolution: this resolution is derived using a  $\square\square$ -rule or a  $\square$ -rule. Then there is a clause  $A$  of the form  $A = \square A'' \vee A'$ , and possibly a "side"-clause  $B = \square B'' \vee B'$ , such that  $A''(, B'') \Rightarrow_{\alpha} I'$ , and thus  $I = \square I' \vee A(\vee B')$ . The  $\beta$ -resolution involves the literal produced in the  $\alpha$ -resolution (we have studied the other case first). Let us start with the case where this  $\beta$ -resolution does not involve any  $\diamond$ -literal: it is a  $\square\square$ - or a  $\square$ -resolution: there may be a clause  $C = \square C'' \vee C'$ , with  $I'(, C'') \Rightarrow_{\beta} F'$ , and  $F = \square F' \vee A'(\vee B') \vee (C')$ . Using an inductive hypothesis again, we can assume there are clauses  $I^*$  and  $F'^*$  such that  $A''(, C'') \Rightarrow_{\beta} I^*$  and  $I^*(, B'') \Rightarrow_{\alpha} F'^*$ . (For the sake of simplicity, we assume that there are no other intermediate clauses  $F_1^*, \dots, F_n^*$ ; the proof would be very similar with more intermediate clauses.) In this case, we can choose  $I^* = \square I^* \vee A'(\vee C')$  and  $F^* = F'^* \vee A'(\vee B')(\vee C')$ :



If the other literal of the  $\beta$ -resolution is a  $\diamond$ -literal, we have  $C = \diamond(C'', E) \vee C'$ , and  $F = \diamond(C'', E, F') \vee A'(\vee B')(\vee C')$ , with  $I', C'' \Rightarrow_{\beta} F'$ . Since  $A''(, B'') \Rightarrow_{\alpha} I'$ , the same inductive hypothesis again allows us to assume that there are clauses  $I^*$  and  $F'^*$  such that  $A'', C'' \Rightarrow_{\beta} I^*$  and  $I^*(, B'') \Rightarrow_{\alpha} F'^*$ . In order to get  $I'$  in the  $\diamond$  literal after the resolution, we need to perform another  $\alpha$ -resolution. In the end, we can choose  $I^* = \diamond(C'', E, I^*) \vee A'(\vee B')$  and  $F^* = \diamond(C'', E, F'^*, I') \vee A'(\vee B')(\vee C')$ :





Let  $P = \{s, p, q\}$ , we now compute  $\text{Supp}(P, R)$ . The clauses  $\Diamond(\neg s, \Diamond q)$ ,  $\Diamond(\neg s, \Diamond q, \Diamond \Box p)$  and  $\Box(\Diamond \Box p \vee s)$  are suppressed by  $\text{Supp}(P, \bullet)$  because they contain only occurrences of  $s$  and  $q$ . The clauses  $\Box(r \vee \Box \Diamond \neg p)$  and  $\Box(r \vee s)$  are discarded too, because they do not specify in which case  $r$  is true. In the clause  $\Diamond(\neg s, \Diamond q, \Diamond \Box p, r)$ , one conjunct is kept,  $r$ . So  $R^{\text{no}P} = \{\Diamond r\}$ .

## 4 Uniform interpolation by resolution

Suppose now that we want to compute a uniform interpolant for clause set  $S$  and a sublanguage  $\mathcal{L}^{\text{no}P}$  defined by a subset of variables  $P$ :  $\mathcal{L}^{\text{no}P}$  is the sublanguage of  $\mathcal{L}$  whose formulas have no occurrences of atoms in  $P$ . We can proceed as follows:

1. recursively add to  $S$  all clauses obtained from  $S$  by resolutions on variables from  $P$ : this gives a set of clauses  $S^{\text{res}(P)}$ ; then
2. suppress from  $S^{\text{res}(P)}$  all information about variables from  $P$ ; formally, we define an operator  $\text{Supp}$  such that  $\text{Supp}(P, C)$  associates to clause  $C$  a clause that “forgets” what  $C$  says about variables from  $P$ :
  - if  $C$  is of the form  $p \vee C'$  or  $\neg p \vee C'$  for some  $p \in P$ , then  $\text{Supp}(P, C) = \top$ ;
  - otherwise, that is, if no variable of  $P$  appears at “ground” level in  $C$ :  $\text{Supp}(P, C' \vee \Diamond R_1 \vee \dots \vee \Diamond R_n \vee \Box C_1 \vee \dots \vee \Box C_n) = C' \vee \Diamond \text{Supp}(P, R_1) \vee \dots \vee \Diamond \text{Supp}(P, R_n) \vee \Box \text{Supp}(P, C_1) \vee \dots \vee \Box \text{Supp}(P, C_n)$  where for each  $i$ ,  $\text{Supp}(P, R_i) = \{\text{Supp}(P, C) \mid C \in R_i\}$ . We also perform the natural simplification:  $\Box \top \equiv C \vee \top \equiv \top$  and  $R \cup \{\top\} \equiv R$ .

Let  $S^{\text{no}P} = \{\text{Supp}(P, C) \mid C \in S^{\text{res}(P)}\}$ . We claim that  $S^{\text{no}P}$  is a uniform interpolant of  $S$  on  $\mathcal{L}^{\text{no}P}$ . In order to see this, suppose that  $\phi$  is a formula of  $\mathcal{L}^{\text{no}P}$  such that  $S \models \phi$ : let  $S'$  be a conjunctive normal form of  $\neg\phi$ , by refutation completeness of Enjalbert and Fariñ’as’ resolution system, there is a deduction by resolution of  $\perp$  from  $S \cup S'$ . Let us use Prop. 1 and re-arrange this deduction to get a sequence of inferences where all resolutions on variables from  $P$  are before the others: it has the form  $I_1, \dots, I_k, \dots, I_n$ , where inferences  $I_1, \dots, I_k$  are on variables from  $P$ , and  $I_{k+1}, \dots, I_n$  are resolutions on  $\perp$  or on other variables. Let  $C_1, \dots, C_m$  be the premisses of the inferences  $I_{k+1}, \dots, I_n$  that are in  $S$  or that are consequents of inferences  $I_1, \dots, I_k$ ; and let  $R$  be the set of clauses obtained by applying  $\text{Supp}(P, \bullet)$  to  $C_1, \dots, C_m$ :  $R \subseteq S^{\text{no}P}$ , and there is a deduction by resolution from  $R \cup S'$  for  $\perp$ , since there is a deduction by resolution from  $\{C_1, \dots, C_m\} \cup S'$  and clauses in  $R$  are “simpler” than  $C_1, \dots, C_m$ . Thus  $S^{\text{no}P} \cup S' \models \perp$ , hence  $S^{\text{no}P} \models \phi$ .

**Example 2 (continued)** Suppose we need to compute a  $\mathcal{L}_r$  interpolant of  $R = \{\Box(r \vee \Box \Diamond \neg p), \Box(\Diamond \Box p \vee s), \Diamond(\neg s, \Diamond q)\}$  where  $\mathcal{L}_r$  is the language whose only variable is  $r$ . We first perform all possible resolutions that do not involve  $r$ :

## 5 Conclusion

The results above show that, although it is more complicated than in propositional logic, resolution can be used to compute a uniform interpolant in modal logic  $K$ . In order to have a practical method, one would need to precise a procedure to systematically compute  $P$ -resolvants of a given set of clauses. An algorithm like [2]’s saturation by set could be used, coupled with the elimination of subsumed clauses. From an implementation point of view, an important difference between resolution in propositional logic and in modal logic is the representation of clauses: they can be efficiently represented in a table when there are no modalities. From this point of view, the approach of [1] seems promising: they “flatten” modal formulas, using a naming scheme for the possible worlds, a little like skolemization in first-order logic, and then perform resolution on flat clauses. In order to properly use this approach for interpolation, one would need to define a sort of “de-skolemization” in order to regain modal formulas after interpolation has been performed on flat clauses.

## REFERENCES

- [1] Carlos Areces, Maarten de Rijke, and Hans de Nivelle, ‘Resolution in modal, description and hybrid logic’, *Journal of Logic and Computation*, **11**(5), 717–736, (2001).
- [2] Philippe Besnard, René Quiniou, and Patrice Quinton, ‘A theorem prover for a decidable subset of default logic’, in *Proceedings of the Third National Conference on Artificial Intelligence*, pp. 27–30. Morgan Kaufmann, (1983).
- [3] Meghyn Bienvenu, ‘Prime Implicates and Prime Implicants in Modal Logic’, in *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI’07)*, eds., Robert C. Holte and Adele Howe, pp. 379–384. AAAI Press, (2007).
- [4] Giovanna D’Agostino and Giacomo Lenzi, ‘On modal mu-calculus with explicit interpolants’, *Journal of Applied Logic*, **4**(3), 256–278, (2006).
- [5] Patrice Enjalbert and Luis Fariñas del Cerro, ‘Modal resolution in clausal form’, *Theoretical Computer Science*, **65**(1), 1–33, (1989).
- [6] Silvio Ghilardi and Marek W. Zawadowski, ‘Undecidability of propositional quantifiers in the modal system  $s4$ ’, *Studia Logica*, **55**(2), 259–271, (1995).
- [7] Silvio Ghilardi and Marek W. Zawadowski, *Sheaves, Games, and Model Completions*, volume 14 of *Trends in Logic*, Kluwer, 2002.
- [8] Jürg Kohlas, Serafín Moral, and Rolf Haenni, ‘Propositional Information Systems’, *Journal of Logic and Computation*, **9**(5), 651–681, (1999).
- [9] Jérôme Lang and Pierre Marquis, ‘Resolving inconsistencies by variable forgetting’, in *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR-02)*, eds., Dieter Fensel, Fausto Giunchiglia, Deborah L. McGuinness, and Mary-Anne Williams, pp. 239–250. Morgan Kaufmann, (2002).

- [10] Balder ten Cate, Willem Conradie, Maarten Marx, and Yde Venema, 'Definitorially complete description logics', in *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (kr'06)*, eds., Patrick Doherty, John Mylopoulos, and Christopher A. Welty, pp. 79–89. AAAI Press, (2006).
- [11] Albert Visser, 'Bisimulations, model descriptions and propositional quantifiers', in *Gödel'96: Logical foundations of mathematics, computer science and physics – Kurt Gödel's legacy*, ed., Petr Hájek, A K Peters Ltd, (2001).