LIETUVOS MATEMATIKOS RINKINYS. LMD DARBAI
52 tomas, 2011, 253–256

ISSN 0132-2818
www.mii.lt/LMR/

Resolution for hybrid logics

Stanislovas Norgėla

Faculty of Mathematics and Informatics, Vilnius University
Naugarduko 24, LT-03225, Vilnius
E-mail: stasys.norgela@mif.vu.lt

Abstract. We describe a resolution method and a procedure to transform formulae of some pure hybrid logics into their clausal form.

Keywords: hybrid logic, resolution, clause.

Introduction

Before applying the resolution method to the formulas of classical logic we transform them into clausal form. Well-known transformation methods of the formulas of classical logic are not suitable for neither modal nor hybrid logics. Transformation of formulas of hybrid logics needs a different approach. In [6, 7] Mints et al. describe transformation of formulae into their clausal form for modal logics $S4$ and $S5$. A modal literal is defined as formula of the form l, $\Box l$ or $\Diamond l$, where l is a propositional literal. A modal clause is a disjunction of modal literals. We prove that for every modal logic formula F there exist clauses $D_1, \ldots, D_n$ and a propositional literal l such that sequent $\vdash F$ is derivable in sequent calculus $S4$ (and, accordingly, $S5$) if and only if sequent $\Box D_1, \ldots, \Box D_n, l \vdash$ is derivable. This transformation is the basis for the resolution calculus for modal logic $S4$ presented in [7]. F is a tautology if and only if an empty clause is derivable from the set $\{\Box D_1, \ldots, \Box D_n, l\}$. The paper [10] describes a procedure to transform formulae of hybrid logic $\mathcal{H}(@)$ over transitive and reflexive frames into their clausal form. This paper shows how we can transform formulas of hybrid logics $\mathcal{H}(@,\downarrow)$, $\mathcal{H}(@,\exists)$, $\mathcal{H}(@,E)$ into their clausal form. We also describe resolution method for the hybrid logics $\mathcal{H}(@,\downarrow)$, $\mathcal{H}(@,E)$. For more information about hybrid logic and its properties see [2, 3, 4, 5].

1 Transformation

A *literal* of hybrid logic $\mathcal{H}(@)$ is a formula of the form $l, \Box l, \Diamond l, @_i l$ (where l is a propositional variable, nominal or their negation; i – is nominal). In addition, $\forall xl$, $\exists xl$ are also literals in the logic $\mathcal{H}(@,\exists)$, Al, El – in the logic $\mathcal{H}(@,E)$, $\downarrow x.l$ – in the logic $\mathcal{H}(@,\downarrow)$.

A *clause* is a formula of the form L, $\Box L$, $@_i L$ (where L is a disjunction of hybrid literals). In addition, a formula of the form $@_i \forall xL$ is clause of logic $\mathcal{H}(@,\exists)$ and a formula $@_i \downarrow x.L$ is clause of logic $\mathcal{H}(@,\downarrow)$.

First of all, consider a formula of logic $\mathcal{H}(@)$. We transform a subformula of the form $@_i G$ into clause $@_i \neg G \vee a$ (where a is a new propositional variable). We transform other subformulas similarly as described in [2, 3, 10]. If any subformula occur in the scope of modal operators $\Box, \Diamond$ we write $@_z$ in the front. In the following we assume that the variable z is *reserved*, that is, z does not occur in the formulas under consideration.

Assume we want to know whether a sequent $\vdash F$ has a deduction in the sequent calculus H presented in [9, 8]. We denote all the possible subformulas, with the exception of nominals, by the new propositional variables $a, b, c, \ldots$. To the goal formula F assign letter a. Assume the formula F is in negation normal form, that is, the formula contains logical connectives only from the list $\neg, \vee, \wedge$ and the negation symbol appears only in front of nominals. Recall that in this paper we transform only formulas of pure hybrid logic. Obtainable after transformation clauses contain propositional variables and nominals. Formula F has one of the forms: 1. $G \wedge H$, 2. $G \vee H$, 3. $\Box H$, 4. $\Diamond H$, 5. $@_i H$. Suppose the variable b assigned to the formulas G, $\Box H$, $\Diamond H$, $@_i H$ and variable c assigned to formula H.

In the first case formula is derivable if and only if $\neg a, a \vee \neg b \vee \neg c \vdash$ is derivable. In the second case if sequent $\neg a, a \vee \neg b, a \vee \neg c \vdash$ is derivable.

We add new rule

$$(@_z) \quad \frac{\Gamma, @_i G, @_z(u \vee G)}{\Gamma, @_z(u \vee G), @_i \neg u}$$

to the calculus H and denote them by H'. In the third case $\vdash F$ is derivable in the calculus H iff $\neg a, @_z(a \vee \Diamond \neg b) \vdash$ is derivable in calculus H'. Where u is a nominal. In addition, we can apply only the rule $@_z$ to the formulas beginnig with $@_z$.

In the fourth case, if sequent $\neg a, @_z(a \vee \Box \neg b) \vdash$ is derivable in calculus H'. In the fifth case, if sequent $\neg a, a \vee @_i \neg b) \vdash$ is derivable.

We apply the transformation to subformulas b, c and its components as long as it possible. We will say that list of obtained clauses $D_1, \ldots, D_n$ corresponds to formula F. We get the following result.

Theorem 1. *For any formula F of logic $\mathcal{H}(@)$ a sequent $\vdash F$ is derivable in H iff a sequent $D_1, \ldots, D_n \vdash$ corresponding to F is derivable in H'.*

Example 1. $F = \Box\Diamond(i \vee j) \wedge \Diamond\neg j$.

Let the letters e, c, f, d, b, a denote the subformula of, respectively $\neg j$, $\Diamond\neg j$, $i \vee j$, $\Diamond(i \vee j)$, $\Box\Diamond(i \vee j)$, F.

The following list of clauses corresponds to formula F: $\neg a$, $a \vee \neg b \vee \neg c$, $@_z(b \vee \Diamond\neg d)$, $@_z(d \vee \Box\neg f)$, $@_z(f \vee \neg i)$, $@_z(f \vee \neg j)$, $@_z(c \vee \Box\neg e)$, $@_z(e \vee j)$.

The case of logic $\mathcal{H}(@, E)$ is treated in a similar manner.

Example 2. $F = \Box E\Diamond(i \vee j) \wedge A\Diamond\neg j$. Let the letters e, c, c', f, d, d', b, a denote the subformula of, respectively $\neg j$, $\Diamond\neg j$, $A\Diamond\neg j$, $i \vee j$, $\Diamond(i \vee j)$, $E\Diamond(i \vee j)$, $\Box E\Diamond(i \vee j)$, F.

The following list of clauses corresponds to formula F: $\neg a$, $a \vee \neg b \vee \neg c'$, $@_z(b \vee \Diamond\neg d')$, $@_z(d' \vee A\neg d)$, $@_z(d \vee \Box\neg f)$, $@_z(f \vee \neg i)$, $@_z(f \vee \neg j)$, $@_z(c' \vee E\neg c)$, $@_z(c \vee \Box\neg e)$, $@_z(e \vee j)$.

The new propositional variables become in logics $\mathcal{H}(@, \exists)$, $\mathcal{H}(@, \downarrow)$ functions of nominal variables.

Examples.

1. Logic $\mathcal{H}(@,\exists)$. We transform formula $\Box\exists x\Diamond(x\vee j)\wedge\forall y\Diamond\neg y$.

We introduce new variables a,b,c,d,e,f,c',d' (where z is reserved) to get a set of clauses.

Consider the following list of subformulas: $\neg y$, $\Diamond\neg y$, $\forall y\Diamond\neg y$, $x\vee j$, $\Diamond(x\vee j)$, $\exists x\Diamond(x\vee j)$, $\Box\exists x\Diamond(i\vee j)$, F and denote them respectively by e,c,c',f,d,d',b,a.

The following list of clauses corresponds to formula F: $\neg a$, $a\vee\neg b\vee\neg c'$, $@_z(b\vee\Diamond\neg d')$, $@_z(d'\vee\forall x\neg d(x))$, $@_z\forall x(d(x)\vee\Box\neg f(x))$, $@_z\forall x(f(x)\vee\neg x)$, $@_z\forall x(f(x)\vee\neg j)$, $@_z(c'\vee\exists y\neg c(y))$, $@_z\forall y(c(y)\vee\Box\neg e(y))$, $@_z\forall y(e(y)\vee y)$.

2. Logic $\mathcal{H}(@,\downarrow)$. We transform formula $\Box\downarrow x.\Diamond(x\vee j)\wedge\Diamond\neg j$.

We introduce new variables *a, b, c, d, e, f,* d' (where z is reserved).

We have the following list of subformulas: $\neg j$, $\Diamond\neg j$, $x\vee j$, $\Diamond(x\vee j)$, $\downarrow x.\Diamond(x\vee j)$, $\Box\downarrow x.\Diamond(i\vee j)$, F and denote them respectively by e,c,f,d,d',b,a.

The following list of clauses corresponds to formula F: $\neg a$, $a\vee\neg b\vee\neg c$, $@_z(b\vee\Diamond\neg d')$, $@_z(d'\vee\downarrow x.\neg d(x))$, $@_z\downarrow x.(d(x)\vee\Box\neg f(x))$, $@_z\downarrow x(f(x)\vee\neg x)$, $@_z\downarrow x(f(x)\vee\neg j)$, $@_z(c\vee\Box\neg e))$, $@_z(e\vee j)$.

2 Resolution method

We describe the rules of the resolution method for logics $\mathcal{H}(@)$, $\mathcal{H}(@,\downarrow)$ and $\mathcal{H}(@,E)$. We do not write the rules of the resolution method for a logic $\mathcal{H}(@,\exists)$ because we do not know skolemization in this logic. Rules for logic $\mathcal{H}(@)$.

z is reserved, n is new.

$$\frac{@_s(F\vee G)}{@_sF\vee @_sG}\quad \frac{@_si\vee H,\quad @_s\neg i\vee G}{H\vee G}\quad \frac{@_z(u\vee G),\quad @_i\neg u}{@_iG}$$

$$\frac{@_s\Diamond G\vee H}{@_s\Diamond n\vee H,\quad @_nG\vee H}\quad \frac{@_s\Box G\vee H,\quad @_s\Diamond i\vee F}{@_iG\vee H\vee F}\quad \frac{@_i\Diamond(G\vee H)}{@_i\Diamond n,\quad @_n(G\vee H)}$$

$$\frac{@_s@_iG\vee H}{@_iG\vee H}\quad \frac{@_si\vee G,\quad F(i)\vee H}{F(s)\vee H}\quad \frac{@_si\vee G}{@_is\vee G}\quad \frac{@_s\neg s\vee G}{G}$$

The logic $\mathcal{H}(@,\downarrow)$ also contains a rule:

$$\frac{@_s\downarrow x.G(x)\vee H}{@_sG(s)\vee H}.$$

Logic $\mathcal{H}(@,E)$ contains two new rules:

$$\frac{@_sEF\vee H}{@_nF\vee H}\qquad \frac{@_sAF\vee H}{@_iF\vee H}.$$

Theorem 2. *A formula $\neg F$ of logics $\mathcal{H}(@,\downarrow)$, $\mathcal{H}(@,E)$ is derivable in sequent calculus H iff empty clause is derivable from set of clauses corresponding to formula F.*

Proof. The described rules are an adaptation of the resolution method [1] to the clauses under consideration. □

Conclusions

The described transformation produces clauses of very simple form. Resolution method become much simpler.

References

[1] C. Areces, H. de Nivelle and M. de Rijke. Resolution in modal, description and hybrid logic. *J. Logic Comput.*, **11**(5):717–736, 2001.

[2] C. Areces and B. ten Cate. Hybrid logics. In *Handbook of Modal Logic*, pp. 821–868. Elsevier, 2006.

[3] P. Blackburn and J. Seligman. Hybrid languages. *J. Logic, Lang. Inf.*, **4**:251–272, 1995.

[4] T. Bolander and T. Braüner. Two tableau-based procedures for hybrid logic. In *Methods for Modalities, vol. 4*, pp. 79–96. Adlershof, 2005.

[5] T. Braüner. Natural deduction for hybrid logic. *J. Logic, Comput.*, **14**(3):329–353, 2004.

[6] G. Mints. Gentzen-type systems and resolution rule. Part I. *Lect. Not. Comput. Sci.*, **417**:198–231, 1988.

[7] G. Mints, V. Orevkov and T. Tammet. Transfer of sequent calculus strategies to resolution for S4. In *Proof Theory and Modal Logic*, pp. 79–96. Kluwer Academic Publishers, 1996.

[8] S. Norgėla ir A. Šalaviejienė. Sequent calculus for hybrid logic. *Liet. mat. rink.* (spec. nr.), pp. 238–241, 2006.

[9] S. Norgėla ir A. Šalaviejienė. Some decidable classes of formulas of pure hybrid logic. *Liet. mat. rink.*, **47**(4):563–572, 2007.

[10] S. Norgėla ir L. Petrauskas. Transformations of formulae of hybrid logic. *Liet. mat. rink., LMD darbai*, **51**:342–346, 2010.

REZIUMĖ

Rezoliucijų metodas hibridinėms logikoms
S. Norgėla

Aprašytas grynųjų hibridinių logikų formulių transformavimo į disjunktų aibę algoritmas bei rezoliucijų metodas, kuris taikomas disjunktų aibėms.

Raktiniai žodžiai: hibridinė logika, rezoliucijų metodas, disjunktas.