

Past-Future Separation and Normal Forms in Temporal Predicate Logic Specifications

Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
treur@cs.vu.nl <http://www.cs.vu.nl/~treur>

Abstract

Temporal specifications are often used when phenomena are modelled where dynamics play a main role. If simulation is one of the aims of modelling, usually a restricted, executable modelling language format is used, based on some form of past to future implications. In this paper a detailed transformation procedure is described that takes any temporal predicate logic specification and generates a specification in a past implies future normal format. The procedure works for temporal specifications in which the atoms either express time ordering relations or are state-related, i.e., include only one time variable.

1 Introduction

For many phenomena, to analyse and model them the temporal aspect is a dominant factor to be taken into account. To model such phenomena, often some form of formal temporal specification is used. A number of temporal modelling languages are available, varying from Dynamical Systems Theory based on differential equations (e.g., Ashby, 1952; Kelso, 1995; Port and van Gelder, 1995), to temporal logical languages (e.g., Gabbay, 1989; Gabbay, Hodkinson, and Reynolds, 1994; Fisher, 1997, 2005). From the logical side, Galton (2003) distinguishes two main streams: modal logic approaches to temporal logic, and predicate logic approaches to temporal logic. For the former stream, see, for example, (Fisher, 1997, 2005; Gabbay, 1989; Gabbay, Hodkinson, Reynolds, 1994). In (Galton, 2006) he addresses different types of approaches in the latter stream in more detail.

As far as expressivity is concerned, such languages to address dynamics show much variation; for example, purely quantitative approaches based on numerical mathematical formulae, versus purely qualitative approaches based on propositional logical formulae. Both of these extreme cases have limitations for practical usability. For realistic practical applications, predicate logical expressivity is

useful to express qualitative aspects, using variables, and numerical expressions for quantitative relations sometimes have to be integrated with such logical expressions. For such reasons within AI often predicate logic approaches to dynamics are considered, as also is remarked by Galton (2003). Examples of such predicate logical temporal languages are situation calculus (McCarthy and Hayes, 1969; Levesque, Pirri, and Reiter, 1998; Reiter, 2001), event calculus (Kowalski and Sergot, 1986; Kowalski and Sadri, 1997), and the Temporal Trace Language TTL (Jonker and Treur, 2002; Bosse, Jonker, Meij, Sharpanskykh and Treur, 2009); see also (Vila and Reichgelt, 1996).

A theme that has a long tradition in the area of temporal languages is the use of normal forms for temporal specifications, that usually have some logical structure of the form ‘past implies future’. Such normal forms are suitable for execution, in contrast to temporal specifications with unrestricted format: the notion of Executable Temporal Logic is based on this; e.g. (Gabbay, 1989; Barringer et al., 1996; Fisher, 2005). There are two ways to obtain specifications of this format. One way is to offer modellers a language with just this restricted format (e.g., Concurrent MetateM; Barringer et al., 1996), and let them take the effort to write their specifications in this format. As usually modellers manage to do this, the question can be raised whether in principle any temporal specification can be rewritten in such format.

Indeed there is also some history on the theme how to transform a general temporal specification into normal form, by what is called separation of past and future. For example, in (Gabbay, 1989; Gabbay, Hodkinson and Reynolds, 1994; Hodkinson and Reynolds, 2005) contributions to this theme can be found. Most of this literature addresses the modal-logic-based stream of temporal languages. However, in (Gabbay, Hodkinson and Reynolds, 1994; Hodkinson and Reynolds, 2005) also the predicate logic case is addressed. In the latter a nonconstructive, modeltheoretic proof is given for the possibility of separation of any predicate logical temporal formulae. In the former a brief sketch of an inductive proof is given that can be considered constructive. However, in neither of the references a more detailed procedure for transformation of a general predicate logic temporal formula into a separated normal form is given. The current paper fills this omission by providing a detailed procedure to transform a predicate logical temporal formula in a separated format in the form of a set of ‘past implies future’ formulae.

In this paper, first in Section 2 the type of languages covered are discussed. Section 3 discusses the ideas of separation and normal form and the type of format that is aimed for in this paper. Next, in Sections 4 to 7 the transformation procedure is described in the different steps needed. For each step, in the form of a proposition it is formulated (a) what can be achieved, (b) in which specific form, and (c) by which transformation steps this can be obtained. Moreover, the different steps are described independently, so that each step can be (re)used in any situation whenever appropriate, without having to go through the whole procedure. More specifically, Section 4 shows the two steps how to limit scopes of quantifiers and to differentiate the time quantifiers into past and future quantifiers. Section 5 describes the two steps for transformation into prenex conjunctive normal form used, and in Section 6 it is shown how to transform this format into a prenex format with matrix in the form of a conjunction of implications from past to future. In Section 7 the last transformation steps of the procedure is

described, namely how to distribute the quantifiers over and within the past to future implications. Here one of the less easy aspects is solved, namely how to distribute an existential quantifier over a conjunction of implications. In Section 8 an overview of the whole transformation procedure is summarised and the main theorem formulated. Section 9 is a discussion.

2 The Class of Temporal Predicate Logics for which the Approach is Applicable

The approach put forward in this paper applies to more than one specific logic. This section addresses the assumptions made on the class of predicate logical temporal languages for which the approach can be used. Many predicate logic temporal languages fit in this class, including, for example, situation calculus (McCarthy and Hayes, 1969; Levesque, Pirri, and Reiter, 1998; Reiter, 2001), event calculus (Kowalski and Sergot, 1986; Kowalski and Sadri, 1997), and the Temporal Trace Language TTL (Jonker and Treur, 2002; Bosse, Jonker, Meij, Sharpanskykh and Treur, 2009).

To characterise this class, first of all, full first-order expressivity is allowed, with both time and non-time variables and quantifiers over them. However, some restrictions on the atoms have to be made. In case of more than one time variable in an atom mixed situations can arise: one of these variables may refer to the past whereas another one refers to the future. Then classification of this atom as a past atom or a future atom is not possible and hence separation would be impossible.

For example, if R is a relation symbol, then $\forall u_1 < t \exists v_2 \geq t R(a, u_1, v_2)$ is a formula, with an atom $R(a, u_1, v_2)$ where in general past and future cannot be separated, unless more is known about this relation R (e.g., knowledge indicating how to replace this atom by another, equivalent formula). This shows that relations involving different time points can pose problems to the separation. One standard relation between different time points often used in temporal formulae is the binary time ordering relation $<$. However, in this special case indeed more is known about the relation $<$ in the form of ordering axioms, e.g., transitivity. In principle, every relation involving more than one time variable could be handled, if knowledge is available relating mixed expressions involving past and future variable in one atom, to expressions involving atoms with only past variables or only future variables.

For many cases of logical dynamic modelling, state properties p are incorporated, and for each state property it can be expressed that it holds at a certain point in time t , for example by an atom of the form $\text{holds_at}(p, t)$, or by including a time variable in p , for example $p(t)$. On top of such state-related formulae, temporal relationships can be expressed using the standard time ordering relation. This way of modelling provides a neat separation between state properties and time ordering relations, as a kind of modular way of modelling. Having such possibilities already, it can be debated whether other (atomic) relations involving more than one time variable (except the time ordering relations) are useful in modelling dynamic properties. Just using atoms expressing that certain state properties hold at certain time points, and standard order relations between time points as basic elements, in practical cases may provide sufficient expressivity. All other relations may be definable in terms of these basic elements, and it may even be claimed that it provides more insight to model such relations in that modular way. As an example, consider an atom

$\text{is_rising_between}(o_1, \text{temperature}, t_1, t_2)$

expressing that the temperature of object o_1 is rising between time points t_1 and t_2 . This can be defined either by (if it is meant globally)

$\exists c_1, c_2 \text{ holds_at}(\text{has_temperature}(o_1, c_1), t_1) \wedge \text{holds_at}(\text{has_temperature}(o_1, c_2), t_2) \wedge c_1 \leq c_2$

or by (if it is meant locally)

$\forall t, t' [t_1 \leq t \leq t_2 \wedge \text{holds_at}(\text{has_temperature}(o_1, c), t) \wedge \text{holds_at}(\text{has_temperature}(o_1, c'), t') \rightarrow c_1 \leq c_2]$

In many applications the latter way of modelling can be used. To summarize these considerations, both from the practical modelling perspective, and from the perspective of feasibility of separation of past and future at the level of atoms, it seems not too bold to assume that every atom either involves only one time variable, or it concerns a time ordering relation.

Definition (State-Time Partitioned Formula)

- a) An atom involves only one time variable or constant is called *state-related atom*. An atom expressing a time ordering relation is called a *time-ordering atom*.
- b) A formula is called *state-time partitioned* if every atom that occurs in it either is a state-related atom or a time ordering atom.

Examples of state-related atoms are

$\text{holds_at}(\text{has_temperature}(o_1, c_1), t_1)$
 $\text{has_temperature_at}(o_1, c_1, t_1)$

An example of time ordering atom is

$t_1 < t_2$

Based on the above deliberations we make the following basic assumption on the predicate logical language used for the rest of the paper.

Assumption

Every formula is state-time partitioned.

3 On Separation and Past Implies Future Normal Form

In this section an outline is sketched of the problem that is addressed and some of the ingredients of the approach are defined.

Definition 1 (Pure Formula for t)

Let φ be a closed formula, and t a time variable or constant.

- a) The formula φ is a *past formula* for t if each time variable different from t is restricted to the time interval before t . In other words, every time quantifier for a variable s is relativised by a restriction of the form $s < t$.
- b) The formula φ is a *future formula* for t if each time variable different from t is restricted to the time interval after t . In other words, every time quantifier for a variable s is relativised by a restriction of the form $s \geq t$.

c) The formula φ is *pure* for t if it is either a past formula or a future formula for t .

Definition 2 (Separated Formula and Past Implies Future Normal Form)

- a) The formula φ is *separated* if it is a boolean combination of closed pure formulae.
- b) The formula φ is in *past implies future normal form* if it is a conjunction of implications $A \rightarrow B$ where A is a closed past formula and B is a closed future formula.

Notice that in the above definition in a direct manner b) implies a), but not conversely. To obtain a formula satisfying b) from a formula satisfying a), some rewriting work has to be done. In the approach presented in this paper b) is obtained, thus also providing a). The approach is based on a number of steps. First the variables are related to the given t which plays the role of the current time point by differentiation of, for example, $\forall t_1 A$ in $\forall t_1 < t A \wedge \forall t_1 \geq t A$. The resulting formula is rewritten in prenex conjunctive normal form. Each clause in this formula is reorganised in ‘past implies future’ format. For example, a clause $a \vee b \vee c \vee d$, where b and d are past literals and a and c future literals gets the form $(\neg b \wedge \neg d) \rightarrow (a \vee c)$. Finally, the quantifiers are distributed over and within these implications; this is the most elaborate step in the procedure, and will be addressed in detail in Section 7. In some more detail the 8 steps are as follows; here each number corresponds to a proposition in one of the subsequent sections with the same number:

1. Limitation of the scopes of the quantifiers (Section 4)
2. Differentiation of time variables into past and future time variables (Section 4)
3. Elimination of past-future mixed atoms (Section 4)
4. Transformation into prenex normal form (Section 5)
5. Transformation into conjunctive normal form (Section 5)
6. Rewriting a (disjunctive) clause in the form of a past to future implication (Section 6)
7. Limitation of the scopes of the quantifiers (Section 7)
8. Distribution of quantifiers over past to future implications (Section 7)

Each of these steps can have its value in a specific situation; therefore the steps are addressed separately and more or less independently in the subsequent sections. However, taken together, the steps form the basis of an algorithm to transform any state-past partitioned formula into a past implies future normal form.

4 Differentiating Time Variables into Past and Future Time Variables

The idea is that a formula is rewritten into an equivalent one such that time variables that occur in this formula always either are limited (relativized) to past or to future time points with respect to a given time point t . As an example, suppose $\psi(t_1, t_2)$ is a formula in which time variables t_1, t_2 occur. The idea is that different cases in the sense of ordering relation for each of the variables with respect to t are considered: $t_1 < t, t_1 \geq t$ and $t_2 < t, t_2 \geq t$, i.e., in combination four cases: $t_1 < t$ and $t_2 < t, t_1 < t$ and $t_2 \geq t, t_1 \geq t$ and $t_2 < t, t_1 \geq t$ and $t_2 \geq t$. To have different variable names in different parts of the formula, for $t_1 < t$ the variable t_1 is replaced by (*past time variable*) u_i , for $t_1 \geq t$ by (*future time variable*) v_i .

If t_1 and t_2 is any pair of time variables, then as an example the formula $\varphi(t_1, t_2)$ of the form $\forall t_1 \exists t_2 \varphi(t_1, t_2)$ can be rewritten into an equivalent formula by replacing each time variable t_i by past and future time variables u_i and v_i in the following manner:

$$\begin{aligned} \forall t_1 \exists t_2 \varphi(t_1, t_2) &\Leftrightarrow \forall u_1 < t \exists t_2 \varphi(u_1, t_2) \wedge \\ &\quad \forall v_1 \geq t \exists t_2 \varphi(v_1, t_2) \\ &\Leftrightarrow \forall u_1 < t [\exists v_2 < t \varphi(u_1, v_2) \vee \exists v_2 \geq t \varphi(u_1, v_2)] \wedge \\ &\quad \forall v_1 \geq t [\exists u_2 < t \varphi(v_1, u_2) \vee \exists v_2 \geq t \varphi(v_1, v_2)] \end{aligned}$$

This idea is formulated in more precise terms in Proposition 2 below. However, first a transformation is used to get the scope of quantifiers minimal. Note that throughout the paper implicitly commutation and association rules for conjunction and disjunction are assumed, to avoid repeating them in different orders or associations. The same for quantifiers: $\forall x \forall y A \rightarrow \forall y \forall x A$, and $\exists x \exists y A \rightarrow \exists y \exists x A$ can be used whenever appropriate. Note that in Definition 3 and Proposition 1 all (time and non-time) quantifiers are addressed, whereas Proposition 2 focuses on the time quantifiers only.

Definition 3 (Minimal Quantifier Scope)

The occurrence of a quantifier Q for variable x in a formula has a *minimal scope* if it occurs as $Qx [A \text{ c } B]$ where c is a binary logical connective and x occurs in both A and B , or it occurs as $Qx A$ with A an atom in which x occurs.

Proposition 1 (Limitation of Quantifier Scopes)

Let φ be a closed formula. Then the following hold.

- There is a closed formula φ' equivalent to φ in which every quantifier has a minimal scope.
- The transformation from φ to φ' in a) can be obtained by the following transformation steps.

<p><i>general:</i></p> $\begin{aligned} \exists x [\neg A] &\rightarrow \neg \forall x A \\ \forall x [\neg A] &\rightarrow \neg \exists x A \\ \forall x [A \wedge B] &\rightarrow \forall x A \wedge \forall x B \\ \exists x [A \vee B] &\rightarrow \exists x A \vee \exists x B \\ \exists x [A \rightarrow B] &\rightarrow [\forall x A] \rightarrow \exists x B \end{aligned}$	<p><i>when x does not occur in A:</i></p> $\begin{aligned} \exists x [A \wedge B] &\rightarrow A \wedge \exists x B \\ \forall x [A \vee B] &\rightarrow A \vee \forall x B \\ \forall x [A \rightarrow B] &\rightarrow A \rightarrow \forall x B \end{aligned}$ <p><i>when x does not occur in B:</i></p> $\begin{aligned} \forall x [A \rightarrow B] &\rightarrow [\exists x A] \rightarrow B \\ \forall x B &\rightarrow B \\ \exists x B &\rightarrow B \end{aligned}$
---	---

■

Notice that the implicitly assumed quantifier commutation rules can be used in those cases that the innermost quantifier's scope cannot be reduced, but the outermost quantifier's scope can. Application of these transformation rules requires adequate control, of course. Next, in Proposition 2 the differentiation of time quantifiers is addressed. Here, for example, $A[v_i/t_i]$ denotes the formulae A with v_i substituted for t_i . Moreover, it is assumed that the time variables occurring in φ are t_1, t_2, t_3, \dots , and all quantifiers use a different variable name (otherwise, the variables in φ can be renamed first). The transformation rules can be applied, for example, from outside to inside.

Proposition 2 (Differentiation of Time Quantifiers)

Let φ be a closed formula. Then the following hold.

- a) There is a closed formula φ' equivalent to φ in which every time quantifier is ranging either over the past or the future with respect to a given t , and any atom A' occurring in φ' is obtained from an atom A occurring in φ by replacing every time variable in A by another (past or future) time variable. More specifically, any occurrence of a time quantifier in φ' uses a unique variable name u_i or v_j and is of the form $\forall u_i < t$, $\forall v_j \geq t$, $\exists u_i < t$, or $\exists v_j \geq t$.
- b) The transformation from φ to φ' in a) can be obtained by the following transformation steps introducing for any occurring time variable t_i a differentiation into a pair of new time variables: u_i used over the past and v_i used over the future with respect to t .

For any occurrence of a universal quantifier over t_i :

$$\forall t_i A \quad \rightarrow \quad [\forall u_i < t A[u_i/t_i] \wedge \forall v_i \geq t A[v_i/t_i]]$$

For any occurrence of an existential quantifier over t_i :

$$\exists t_i A \quad \rightarrow \quad [\exists u_i < t A[u_i/t_i] \vee \exists v_i \geq t A[v_i/t_i]] \quad \blacksquare$$

Notice that by these transformation rules, for each time quantifier that occurs, the length of the subformula in the scope of this quantifier is doubled. This adds not much complexity in length when quantifiers are most local (lower in the formula nesting tree), and more complexity in length when they are more global (higher in the tree). In the latter, worst case the length of the formula can roughly become multiplied by 2^n where n is the number of time quantifiers. Part of this added complexity is taken away by the next step, namely the elimination of past-future mixed atoms that are created.

Assuming differentiation of time variables into past and future time variables, state-related atoms (in which only one time variable occurs) can be classified in a straightforward manner as a past atom or future atom according to whether this variable is a past or future variable. This applies, for example, to atoms of the form $\text{holds_at}(p, u_i)$ which indicates that state property p holds at time u_i : this is called a *past atom*; similarly $\text{holds_at}(p, v_j)$ is a *future atom*. For non-unary relations, in the special case of the time ordering relation $<$ the ordering axioms are given, e.g., transitivity. So, as the variables u_i are ranging over past and v_j over future time points with respect to t , for example, an atom of the form $u_i < v_j$ is always true due to transitivity: $u_i < t \leq v_j \Rightarrow u_i < v_j$, so it can be replaced by *true*. Similarly, an atom of the form $v_j < u_i$ can be replaced by *false*, because $v_j < u_i$ and $u_i < v_j$ cannot be both true by antisymmetry: $\forall x, y [x < y \rightarrow \neg y < x]$. For the time ordering relations, atoms of the form $u_i < u_j$ or $v_i < v_j$ remain, and they can be classified as past and future atoms, respectively. Moreover, an equality relation such as $u_i = v_j$ can be replaced by *false* as well.

Proposition 3 (Elimination of Past-Future Mixed Atoms)

Let a state-time partitioned formula φ be given such that all time variables are differentiated in past and future time variables with respect to a given t . Then the following hold.

a) The formula φ is equivalent to a formula φ' such that only atoms from φ occur in φ' and all atoms occurring in φ' either involve only past time variables (past atom) or only future time variables (future atom).

If φ has the form of a clause then also φ' can be taken in the form of a clause, with set of literals that is a subset of that of φ .

b) The formula φ' in a) can be obtained by inspecting the atoms occurring in φ . Classify an atom as past atom if only past variables occur in it and as future atom if only future variables occur. Those atoms that are mixed (containing both a past and a future variable) are eliminated by the following transformation rules:

$u_i = v_j \rightarrow \text{false}$	$\text{true} \wedge A \rightarrow A$	$\text{false} \wedge A \rightarrow \text{false}$
$v_j < u_i \rightarrow \text{false}$	$\text{true} \vee A \rightarrow \text{true}$	$\text{false} \vee A \rightarrow A$
$u_i < v_j \rightarrow \text{true}$	$\text{true} \rightarrow A \rightarrow A$	$\text{false} \rightarrow A \rightarrow \text{true}$
$\neg \text{true} \rightarrow \text{false}$	$A \rightarrow \text{true} \rightarrow \text{true}$	$A \rightarrow \text{false} \rightarrow \neg A$
$\neg \text{false} \rightarrow \text{true}$	$\forall x \text{ true} \rightarrow \text{true}$	$\forall x \text{ false} \rightarrow \text{false}$
	$\exists x \text{ true} \rightarrow \text{true}$	$\exists x \text{ false} \rightarrow \text{false}$

■

Notice, for the case φ has conjunctive normal form, i.e., it has the form of a conjunction of clauses, consisting of disjuncts (which are literals: either atoms or negations of atoms), if by these transformation rules a false or a $\neg \text{true}$ occurs as one of the disjuncts in a clause, this disjunct can be left out of the clause; if a true or a $\neg \text{false}$ occurs as one of the disjuncts this clause, the whole clause can be replaced by true and hence be left out of the conjunction. In this sense, Proposition 3 often entails simplification.

Whether or not many of these past-future mixed atoms occur after application of Proposition 3, just depends on the initial formula. Two extreme cases are:

- the formula does not contain any atom expressing a time ordering relation; then no past-future mixed atoms occur as a result from Proposition 2, so there is no simplification possible by Proposition 3.
- every atom in the formula expresses a time ordering relation; then half of the atoms resulting from Proposition 2 will be past-future mixed and can be eliminated by Proposition 3, thus decreasing complexity (that was increased by Proposition 2) again back to the level of the original formula

These extreme cases are possible but not realistic. In practical situations it will often be between these extremes: a certain fraction of the atoms will concern time ordering relations and the rest will concern other, state-related atoms with one time element in it.

5 Transformation into Prenex Conjunctive Normal Form

By addressing the variables as in Section 4, in the resulting formulae such as $\varphi(u_1, v_2)$ and $\varphi(v_1, u_2)$, both past and future variables may occur. It was already achieved that the set of atoms occurring in the formula can be partitioned in past and future atoms. To separate past and future for the whole formula it is needed to go in further detail into the structure the formulae, until the specific atoms within it are reached. To do so, it is convenient to rewrite the formula first in a standard format. By a wellknown transformation any predicate logic formula can be rewritten into an equivalent formula in a standard format where (1) the occurring quantifiers are the outermost connectives (*prenex normal form*), and (2) the remaining quantifier-free formula (called the *matrix*) has the form of as conjunction of clauses, where each clause is a disjunction of literals, i.e. atoms or negations of atoms (*conjunctive normal form*). See, for example, <http://cs.wvc.edu/~aabyan/Logic/normal.html> for a description and software to perform such transformations. For the case considered, the first step concerns handling the quantifiers, as formulated in Proposition 4 below. Notice that quantifiers addressed here are both time quantifiers and non-time quantifiers.

Definition 4 (Relativised Variables)

A quantifier Q_x with variable x is a shorthand for either a non-relativised or a relativised quantifier $Q_{u_i < t}$ or $Q_{v_i \geq t}$ where u_i respectively v_i is the source variable. When it is written $A[a/x]$ it is meant $A[a/u_i]$, respectively $A[a/v_i]$, denoting the formulae A with a substituted for u_i resp. v_i .

Proposition 4 (Transformation into Prenex Normal Form)

Let φ be a closed formula. Then the following hold.

- There is a closed formula φ' equivalent to φ in which the quantifiers are the outermost connectives.
- The formula φ' in a) can be obtained by the following transformation rules for (relativized) variables x :

$$\begin{array}{ll}
 A \wedge \forall x B & \rightarrow \forall x [A \wedge B] & A \wedge \exists x B & \rightarrow \exists x [A \wedge B] \\
 A \vee \forall x B & \rightarrow \forall x [A \vee B] & A \vee \exists x B & \rightarrow \exists x [A \vee B] \\
 \neg \forall x A & \rightarrow \exists x [\neg A] & \neg \exists x A & \rightarrow \forall x [\neg A] \\
 A \rightarrow \forall x B & \rightarrow \forall x [A \rightarrow B] & A \rightarrow \exists x B & \rightarrow \exists x [A \rightarrow B] \\
 [\forall x A] \rightarrow B & \rightarrow \exists x [A \rightarrow B] & [\exists x A] \rightarrow B & \rightarrow \forall x [A \rightarrow B] \quad \blacksquare
 \end{array}$$

Note that these transformations do not increase the length of the formula. Moreover, notice that Skolemisation is not used in this paper, since in general it would introduce atoms with more than one time variable: if in the prenex form an existential quantifier occurs within the scope of at least one past and one future variable, the Skolem function for this existential quantifier involves these time variables, and hence any atom in which it occurs is a past-future mixed atom which is no state-related atom, nor a time ordering atom, and cannot be classified as either a past or a future atom. Therefore a different way to handle quantifiers is used as will be shown later, in Section 7.

Next, the standard way to transform a quantifier-free formula into Conjunctive Normal Form is as follows.

Proposition 5 (Transformation into Conjunctive Normal Form)

Let φ be a quantifier-free formula. Then the following hold.

- a) There is an formula φ' equivalent to φ , with the same atoms occurring in φ and φ' , where φ' is a conjunction of clauses; here a clause is a disjunction of literals (atoms or negations of atoms).
- b) The formula φ' in a) can be obtained by first eliminating implications, and move negations down to the atoms, and then distribute the \vee over the \wedge , according the following transformation rules:

$A \rightarrow B$	$\rightarrow \neg A \vee B$	$\neg\neg A$	$\rightarrow A$	
$\neg(A \vee B)$	$\rightarrow \neg A \wedge \neg B$	$A \vee B \vee A$	$\rightarrow A \vee B$	
$\neg(A \wedge B)$	$\rightarrow \neg A \vee \neg B$	$A \vee B \vee \neg A$	$\rightarrow \text{true}$	
$A \vee (B \wedge C)$	$\rightarrow (A \vee B) \wedge (A \vee C)$	$A \vee \text{true}$	$\rightarrow \text{true}$	
		$A \wedge \text{true}$	$\rightarrow A$	■

Notice that these transformations can increase the length of the formula (in particular by the last rule of the left column), as is wellknown for the transformation process into conjunctive normal form. The worst case is exponential in length, as is shown by the following wellknown type of example with mn atoms $A_{i,j} (i \in I, j \in J)$, where $\#(I) = m$ and $\#(J) = n$: the formula

$$\bigvee_{i \in I} (\bigwedge_{j \in J} A_{i,j})$$

with length in number of atom occurrences mn transforms into conjunctive normal form as follows:

$$\bigwedge_{j(1), \dots, j(m) \in J} (\bigvee_{i \in I} A_{i,j(i)})$$

which has length n^m in terms of atom occurrences. The right column in Proposition 5, however, can reduce complexity whenever there is an apparent possibility to do so.

6 Transformation of a Clause into a Past to Future Implication

By the transformation into Conjunctive Normal Form, a transparent view is obtained on the formula and how the atoms are embedded in it. What was not handled yet, is a separation of past and future at the level of the formula by regrouping the atoms within the formula. This will be addressed next, based on a partitioning of the set of atoms occurring in a formula into past atoms and future atoms, as accomplished in Section 4. Given such a partitioning of atoms it is not difficult to show how to rewrite a clause into a past to future implication format: transform a clause C into an implication of the form $A \rightarrow B$ where A is the conjunction of the negations of all past literals in C and B is the disjunction of all future literals in C . Thus a quantifier free φ , after rewriting it into Conjunctive Normal Form, can be transformed into a conjunction of implications from past to future.

Proposition 6 (Transforming a Clause into a Past and Future Implication)

Let a clause φ be given such that every atom in φ is either a past atom or a future atom. Then the following hold.

a) The formula φ is equivalent to a formula φ' such that in φ' only atoms from φ occur, and φ' is of the form

$$P \rightarrow Q$$

where P is a conjunction of past literals and Q is a disjunction of future literals.

b) The formula φ' in a) can be obtained from φ as follows. Suppose φ is the disjunction

$$\bigvee PL_i \quad \vee \quad \bigvee FL_j$$

of the past and future literals PL_i and FL_j . Then use the transformation rule

$$\bigvee PL_i \quad \vee \quad \bigvee FL_j \quad \rightarrow \quad \bigwedge \sim PL_i \rightarrow \bigvee FL_j$$

to obtain φ' from φ , where, if a is an atom, $\sim a = \neg a$, and $\sim \neg a = a$. ■

Notice that this transformation does not add complexity to the length of the formula.

7 Distribution of Quantifiers Over a Conjunction of Implications

The transformations from the previous sections can be used to obtain a formula in prenex normal form with quantifier-free formula (the matrix) which is a conjunction of past to future implications with conjunctions of past literals as antecedents and disjunctions of future literals as consequents. However, the scopes of the quantifiers are global, over the whole matrix formula. This section shows how these quantifiers can be rewritten to quantifiers with a single implication as their scope, and even one step further, to quantifiers with a single antecedent or a single consequent of an implication as their scope. Notice that quantifiers addressed here are both time quantifiers and non-time quantifiers.

The first step to consider is to distribute the quantifiers over the conjunction of implications. For universal quantifiers the process is simple, as they distribute over conjunctions in general:

$\forall x [A \wedge B] \Leftrightarrow [\forall x A] \wedge B$	if x occurs in A but not in B
$\forall x [A \wedge B] \Leftrightarrow A \wedge [\forall x B]$	if x occurs in B but not in A
$\forall x [A \wedge B] \Leftrightarrow [\forall x A] \wedge [\forall x B]$	if x occurs in both A and B

The less simple step is to distribute the existential quantifiers. Only when x occurs in only one of the conjuncts we have

$\exists x [A \wedge B] \Leftrightarrow [\exists x A] \wedge B$	if x occurs in A but not in B
$\exists x [A \wedge B] \Leftrightarrow A \wedge [\exists x B]$	if x occurs in B but not in A

These are essentially the same types of steps made in Section 4 to obtain minimal scopes for the quantifiers of the initial formula.

Proposition 7 (Limitation of Scopes of Quantifiers)

Let φ be given in the form of a conjunction of implications

$$\bigwedge_{i \in I} [A_i \rightarrow B_i]$$

where the set V_1 of variables occurring in the A_i is disjoint from the set V_2 of variables occurring in the B_i . Take $V = V_1 \cup V_2$, and for any y in V take $I(y) = \{i \in I \mid y \text{ occurs in } A_i \rightarrow B_i\}$.

Then the following hold.

a) For any variable $x \in V$ the formula φ is equivalent to a formula of the form

$$\varphi_x \wedge \varphi'_x$$

where the formula φ_x consists of those conjuncts in φ in which x occurs, i.e., φ_x is the formula $\bigwedge_{i \in I(x)} [A_i \rightarrow B_i]$, and φ'_x consists of those conjuncts in φ in which x does not occur, i.e., φ'_x is the formula $\bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i]$.

b) For the formula $\varphi_x \wedge \varphi'_x$ in b) it holds

$$\begin{aligned} \exists x \varphi_x \wedge \varphi'_x &\Leftrightarrow [\exists x \varphi_x] \wedge \varphi'_x \\ \forall x \varphi_x \wedge \varphi'_x &\Leftrightarrow [\forall x \varphi_x] \wedge \varphi'_x \end{aligned}$$

c) The following transformation rules concentrate a quantifier to the subformula in which its variable occurs:

$$\begin{aligned} \exists x \bigwedge_{i \in I} [A_i \rightarrow B_i] &\rightarrow \exists x \bigwedge_{i \in I(x)} [A_i \rightarrow B_i] \wedge \bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i] \\ \forall x \bigwedge_{i \in I} [A_i \rightarrow B_i] &\rightarrow \forall x \bigwedge_{i \in I(x)} [A_i \rightarrow B_i] \wedge \bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i] \\ \exists x [\bigwedge_{i \in I(x)} [A_i \rightarrow B_i] \wedge \bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i]] &\rightarrow \exists x [\bigwedge_{i \in I(x)} [A_i \rightarrow B_i]] \wedge \bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i] \\ \forall x [\bigwedge_{i \in I(x)} [A_i \rightarrow B_i] \wedge \bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i]] &\rightarrow \forall x [\bigwedge_{i \in I(x)} [A_i \rightarrow B_i]] \wedge \bigwedge_{i \in I \setminus I(x)} [A_i \rightarrow B_i] \quad \blacksquare \end{aligned}$$

Notice that this proposition does not add complexity to the formula. In fact, strictly spoken this proposition is not needed, but it makes the whole process more efficient, as it allows to apply the next proposition to the smaller subset of only those conjuncts where a variable x actually occurs.

Based on Proposition 7, universal quantifiers can be written distributed over the conjuncts by the following additional steps:

$$\begin{aligned} \forall x \bigwedge_{i \in I(x)} [A_i \rightarrow B_i] &\rightarrow \bigwedge_{i \in I(x)} \forall x [A_i \rightarrow B_i] \\ \forall x [A_i \rightarrow B_i] &\rightarrow [\exists x A_i] \rightarrow B_i \quad \text{if } x \text{ occurs in } A \text{ but does not occur in } B_i \\ \forall x [A_i \rightarrow B_i] &\rightarrow A_i \rightarrow \forall x B_i \quad \text{if } x \text{ occurs in } B \text{ but does not occur in } A_i \end{aligned}$$

In this way a universal quantifier can be handled without adding complexity. However, for an existential quantifier, the case when x occurs in two or more conjuncts cannot be done in this direct manner, as, for example, $\exists x A \wedge B$ in general is not equivalent to $\exists x A \wedge \exists x B$ when x occurs in both A and B . However, for existential quantifiers a more sophisticated transformation is possible to get them distributed, based on the two lemmas presented below (for an alternative approach, see Appendix B). The idea is as follows. Suppose, as an example, a conjunction of two implications

$$\exists x [[A_1 \rightarrow B_1] \wedge [A_2 \rightarrow B_2]]$$

is given, and the variable x does not occur in A_1 and A_2 but does in both B_1 and B_2 . Suppose in a given model M the following holds:

$$\exists x1 [A_1 \rightarrow B_1] [x1/x] \wedge \exists x2 [A_2 \rightarrow B_2] [x2/x]$$

or, in other words, there is an assignment a for $x1$ and $x2$ given such that

$$M \models [[A_1 \rightarrow B_1[a1/x1/x]] \wedge [A_2 \rightarrow B_2[a2/x2/x]]]$$

There are four possibilities for the truth of the antecedents in M :

$$M \models A_1 \wedge A_2$$

$$M \models \neg A_1 \wedge A_2$$

$$M \models A_1 \wedge \neg A_2$$

$$M \models \neg A_1 \wedge \neg A_2$$

In the second case, the second implication is always true, independent of B_2 , so it holds

$$M \models [[A_1 \rightarrow B_1] \wedge [A_2 \rightarrow B_2]] [a1/x]$$

Similarly in the third case, the first implication is always true, so

$$M \models [[A_1 \rightarrow B_1] \wedge [A_2 \rightarrow B_2]] [a2/x]$$

In the third case the conjunction of implications is always true, independent of B_1 or B_2 . So, in the three cases 2 till 4 it holds

$$M \models \exists x [[A_1 \rightarrow B_1] \wedge [A_2 \rightarrow B_2]]$$

For the remaining first case, where $A_1 \wedge A_2$ holds in M this does not hold in general. However, to also cover this case, an addition can be made to the conjunction of clauses, by considering

$$\exists x1 [A_1 \rightarrow B_1] [x1/x] \wedge \exists x2 [A_2 \rightarrow B_2] [x2/x] \wedge \exists x3 [A_1 \wedge A_2 \rightarrow B_1 \wedge B_2] [x3/x]$$

instead of only the first two implications. In the remaining case where $A_1 \wedge A_2$ holds in M , if the last conjunct is also true in M , then also in this case

$$M \models \exists x [[A_1 \rightarrow B_1] \wedge [A_2 \rightarrow B_2]]$$

will hold. This idea, to extend the conjunction of implications to cover different combinations of the antecedents, in a more generalised form for an arbitrary number of conjuncts, is the basis for the approach taken in the Lemmas below.

Definition 5 (Accumulating)

The family of implications $A_i \rightarrow B_i$, $i \in I$ is called *accumulating* when for all $i, j \in I$ with $i \neq j$ it holds that there exists a k such that $A_i \wedge A_j \models A_k$ and $B_k \models B_i \wedge B_j$. An equivalent formulation is: for every finite nonempty subset $J \subseteq I$ there exists a k such that $\bigwedge_{j \in J} A_j \models A_k$ and $B_k \models \bigwedge_{j \in J} B_j$.

Lemma 1

If the family of implications $A_i \rightarrow B_i$, $i \in I$ is accumulating, and the free variable x does not occur in the A_i , then

$$\exists x \bigwedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{i \in I} \exists x [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{i \in I} [A_i \rightarrow \exists x B_i] \quad \blacksquare$$

Lemma 2

a) For any family of implications $A_i \rightarrow B_i, i \in I$ there exist an accumulating family of formulae $A'_j \rightarrow B'_j, j \in J$ such that

$$\bigwedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{j \in J} [A'_j \rightarrow B'_j]$$

where every A'_j is a conjunction of A_i and every B'_j is a conjunction of B_i and

b) More specifically, the index set J and the family $A'_j, B'_j, j \in J$ in a) can be taken as follows:

$$\begin{aligned} J &= P(I) && (P(I) \text{ is the power set of } I) \\ A'_j &= \bigwedge_{i \in j} A_i \\ B'_j &= \bigwedge_{i \in j} B_i \end{aligned} \quad \blacksquare$$

Proofs of these Lemmas and Proposition 8 can be found in Appendix A.

Proposition 8 (Distributing Quantifiers over Conjunctions of Implications)

Let φ be a formula in the form of a conjunction of implications

$$\bigwedge_{i \in I} [A_i \rightarrow B_i]$$

and let x be a variable occurring in φ . Then the following hold.

a) The formula φ is equivalent to a formula φ' of the form

$$\bigwedge_{j \in J} A'_j \rightarrow B'_j$$

such that if x occurs in one or more of the B_i it holds

$$\exists x \bigwedge_{j \in J} [A'_j \rightarrow B'_j] \Leftrightarrow \bigwedge_{j \in J} \exists x [A'_j \rightarrow B'_j] \Leftrightarrow \bigwedge_{j \in J} [A'_j \rightarrow \exists x B'_j]$$

Moreover, every A'_j is a conjunction of A_i and every B'_j is a conjunction of B_i .

b) The formula φ is equivalent to a formula φ'' of the form

$$\bigwedge_{j \in J} A''_j \rightarrow B''_j$$

such that if the variable x occurs in one or more of the A_i it holds

$$\exists x \bigwedge_{j \in J} [A''_j \rightarrow B''_j] \Leftrightarrow \bigwedge_{j \in J} \exists x [A''_j \rightarrow B''_j] \Leftrightarrow \bigwedge_{j \in J} [\forall x A''_j \rightarrow B''_j]$$

Moreover, every A''_j is a disjunction of A_i and every B''_j is a disjunction of B_i .

c) The formulae φ' and φ'' in a) and b) respectively can be obtained from φ as follows.

$$\begin{aligned} \bigwedge_{i \in I} [A_i \rightarrow B_i] &\rightarrow \bigwedge_{j \in P(I)} [\bigwedge_{i \in j} A_i \rightarrow \bigwedge_{i \in j} B_i] \\ \bigwedge_{i \in I} [A_i \rightarrow B_i] &\rightarrow \bigwedge_{j \in P(I)} [\bigvee_{i \in j} A_i \rightarrow \bigvee_{i \in j} B_i] \end{aligned}$$

The following transformation rules handle existential quantifiers for variables in one or more of the B_i , respectively in one or more of the A_i . Here P denotes taking the power set.

(1) if x occurs in the B_i but does not occur in the A_i :

$$\begin{aligned} \exists x \wedge_{i \in I} [A_i \rightarrow B_i] &\quad \rightarrow \quad \wedge_{j \in P(I)} \exists x [\wedge_{i \in j} A_i \rightarrow \wedge_{i \in j} B_i] \\ \exists x [\wedge_{i \in j} A_i \rightarrow \wedge_{i \in j} B_i] &\quad \rightarrow \quad [\wedge_{i \in j} A_i \rightarrow \exists x \wedge_{i \in j} B_i] \end{aligned}$$

(2) if x occurs in the A_i but does not occur in the B_i :

$$\begin{aligned} \exists x \wedge_{i \in I} [A_i \rightarrow B_i] &\quad \rightarrow \quad \wedge_{j \in P(I)} \exists x [\wedge_{i \in j} A_i \rightarrow \wedge_{i \in j} B_i] \\ \exists x [\wedge_{i \in j} A_i \rightarrow \wedge_{i \in j} B_i] &\quad \rightarrow \quad [\forall x [\wedge_{i \in j} A_i] \rightarrow \wedge_{i \in j} B_i] \end{aligned}$$

d) The following transformation rules handle universal quantifiers for variables in one or more of the B_i , respectively in one or more of the A_i :

(1) if x occurs in the A_i or in the B_i :

$$\forall x \wedge_{i \in I} [A_i \rightarrow B_i] \quad \rightarrow \quad \wedge_{i \in I} \forall x [A_i \rightarrow B_i]$$

(2) if x occurs in the B_i but does not occur in the A_i :

$$\forall x [A_i \rightarrow B_i] \quad \rightarrow \quad A_i \rightarrow \forall x B_i$$

(3) if x occurs in the A_i but does not occur in the B_i :

$$\forall x [A_i \rightarrow B_i] \quad \rightarrow \quad [\exists x A_i] \rightarrow B_i \quad \blacksquare$$

Notice that each existential quantifier adds complexity to the formula in the following manner. If x is an existentially quantified variable occurring in $n(x)$ conjunctions of implications in the formula, then this number of conjuncts of implications becomes $2^{n(x)}$. Moreover, the implications themselves also become more complex as their antecedents and consequents grow by a factor up to $n(x)$. So the growth of the part of the formula addressed is about $\frac{1}{2} n(x) 2^{n(x)} = n(x) 2^{n(x)-1}$. So, the added complexity depends to a large extent in how local the variables are in the conjunctive normal form formula. In an extreme case that a variable would occur in all clauses (which would be a kind of holistic case), then $n(x)$ is in proportion with the length of the conjunctive normal form formula, which already can be exponential in the length of the original formula.

8 The Separation and Normal Form Theorem

In this section, first in the form of the formulation of the separation and normal form theorem the previous sections are summarised, in Section 8.1. Next, in section 8.2 some additional control information is discussed. Finally, in Section 8.3 some considerations about complexity are discussed.

8.1 The Theorem

Based on the eight propositions above the separation theorem can be obtained.

Theorem (Separation in Past Implies Future Normal Form)

For any closed state-time partitioned temporal formula φ the following hold.

- The formula φ is equivalent to a separated formula φ' in past implies future normal form.
- The transformation from φ to φ' can be obtained by the transformation rules to obtain

- Limitation of scopes of quantifiers

2. Past-future differentiation of time variables
3. Elimination of past-future mixed atoms
4. Transformation into prenex normal form
5. Transformation into conjunctive normal form
6. Rewriting a clause in past implies future form
7. Limitation of scopes of quantifiers
8. Distribution of quantifiers over past to future implications

as given in the Propositions 1 to 8. ■

8.2 About Control Strategies for the Transformations

The theorem above describes the sequence of the eight transformations to be followed. The content of each of the transformations is specified in the corresponding propositions; in most cases applying the transformation rules in any order on any applicable part of the formula will give a result. However, for some of the transformations additional control strategy information is useful or even necessary, which in cases of more possibilities specifies which transformation rules to apply first and on which part of the formula. Therefore for each of the transformations a possible control strategy is briefly discussed.

1. *Limitation of scopes of quantifiers*

An appropriate control strategy for this transformation is to work from inside out, starting with the quantifiers that are the lowest in the nesting tree of the formula (or, in other words, that have no other quantifiers in their scope), until no transformation rules can be applied anymore, and then work upwards in the tree.

2. *Past-future differentiation of time variables*

Also here an appropriate control strategy is to work from inside out, starting with the quantifiers that are the lowest in the nesting tree of the formula, and then work upwards in the tree.

3. *Elimination of past-future mixed atoms*

Here any mixed atom can be taken and from there worked upwards.

4. *Transformation into prenex normal form*

Here first the outermost quantifiers can be addressed that are not in the scope of any other quantifier, and then work downward in the tree.

5. *Transformation into conjunctive normal form*

For this transformation, first the implication can be eliminated, next negations can be pushed more inside towards the atoms, and finally were possible the disjunctions can be distributed over conjunctions. In the meantime the simplifications rules can be applied as soon as applicable.

6. *Rewriting a clause in past implies future form*

Here the clauses can be treated one-by-one.

7. *Limitation of scopes of quantifiers*

Here the same strategy as in 1. can be used.

8. *Distribution of quantifiers over past to future implications*

Also here the strategy can be starting from the innermost quantifiers and then working upward.

8.3 About Worst Case Complexity

In the sections above already some comments about complexity of the transformation were made, following each proposition. For an estimation of the worst case complexity of the formula obtained during the transformations depends on a number of aspects:

- the length c of the original formula
- the number n of quantified time variables in the formula

It is not easy to make a realistic estimation of the resulting complexity in its dependence on all of these factors. For an idea of how these factors may play their role, we briefly go through the different transformation steps.

For the time quantifier differentiation (Proposition 1), for each time quantifier that occurs, the length of the subformula in the scope of this quantifier is doubled. In the most negative case, when their scopes are rather global and overlapping, the length of the formula can roughly become multiplied by 2^n where n is the number of time quantifiers. This can be expressed by an upper bound of about $2^n c$.

Transformation into prenex form (Proposition 3) does not change complexity much. Transformation to conjunctive normal form (Proposition 4) may be exponential in the length of the formula, which in the worst case would make the number $2^{(2^n)c}$. Transformation into implication form (Proposition 5) does not change complexity. The same holds for concentrating quantifiers (Proposition 6). As the last step, the distribution of quantifiers within the implications (Proposition 7) adds complexity for the existential ones. As the length of the conjunctive normal form formula already can be exponential in the length of the original formula, in the combined worst case, roughly spoken, this would imply double exponential for the length of the resulting formula compared to the length of the original formula (one for the conjunctive normal form transformation, and one for the distribution of quantifiers over the implications). However, for specific cases, the complexity shown in reality depends on a number of other factors as indicated above, which may suggest that in realistic cases the resulting formula may be less complex; see also some considerations in the Discussion below.

9 Discussion

For many phenomena in the natural and artificial world, to analyse and model them the temporal aspect is a dominant factor to be taken into account. Such phenomena range, for example, from biological via neural and cognitive domains to multiagent organisation domains. Examples are modelling bacterial behaviour, conditioning in lower and higher animals, reasoning, belief-desire-intention modelling and modelling adaptive biological organisms or social organisations. Often such a phenomenon can be conceptualised at different aggregation levels that can be identified. For example, for bacterial behaviour, the level of the bacterium as a whole, the level of its two main subprocesses control and metabolism, the level of subprocesses of these, transcription and translation, respectively, and catabolism, anabolism and transport, and the level of specific chemical reactions within such subprocesses; e.g., (Bosse, Jonker, and Treur, 2007). Modelling such phenomena according to different aggregation levels means that their dynamics is described by specifications at these different

levels, connected by logical interlevel relations expressing, for example, that a certain specification at one aggregation level is entailed by a specification at the next lower level; e.g., (Sharpanskykh and Treur, 2006).

Within Computer Science or software-directed areas within Artificial Intelligence, this situation corresponds to specifying behavioural requirements of an overall process that are refined to more specific behavioural requirements for subprocesses at lower levels; e.g., (Herlea Damian, Jonker, Treur, and Wijngaards, 2005). From the viewpoint of verification, the notion of compositional verification is relevant here, where properties for a higher aggregation level are verified by relating them (via interlevel relations) to properties of components at a lower level; e.g., (Roever, Langmaack, and Pnueli, 1998; Jonker and Treur, 2002). In this case the interlevel relations are taken of a simple, self-evident form, or they can be subject of verification themselves; e.g., (Sharpanskykh and Treur, 2006).

Temporal specifications are often used when phenomena are modelled where dynamics play a main role, as in the types of situations as sketched above. If simulation is one of the aims of modelling, usually a restricted, executable modelling language format is used, based on some form of past to future implications. But also for verification a simple format may be needed. For example, if a model checker is used to verify interlevel relations, as in (Sharpanskykh and Treur, 2006), for the specification at the lower aggregation level a format is needed that can be translated easily in the transition system format required for the system description in model checkers; the executable format is such a format; e.g., (Dixon, Bolotov, and Fisher, 2005; Sharpanskykh and Treur, 2006). If a number of (hierarchical) interlevel relations are to be verified in this way, this would require that all of these aggregation levels, except the top level are to be specified in a simple executable format.

In this paper a detailed transformation procedure is described that takes any temporal predicate logic specification and generates a specification in a past implies future normal format. For each step, in the form of a proposition it was formulated (a) what is achieved, (b) in which specific form, and (c) by which transformation steps this can be obtained. Moreover, the different steps are described independently, so that each step can be (re)used in any situation whenever appropriate. The procedure works for any state-time partitioned predicate logic temporal specifications: specifications in which the atoms either express time ordering relations or are state-related, i.e., include only one time variable. Availability of this transformation procedure allows a modeller to avoid specifying in a limited executable format. Temporal specifications of all aggregation levels can have a general form, and the procedure can be used to do simulation or verification.

In Section 8.3 above already the worst case complexity of the transformation was discussed, depending on the length c of the initial formula and the number of time quantifiers n . However, a realistic estimation the complexity of the formula obtained during the transformations depends on a larger number of aspects, in addition to these two factors n and c :

- the proportion between state-related atoms and time-ordering atoms in the formula

- the extent to which the scope of the time quantifiers is more local vs more global in the formula
- the proportion of boolean connectives in the formula versus quantifiers
- the extent to which the distribution of the variables over the clauses resulting from the transformation to conjunctive normal form is more local vs more global

For an idea of how these factors may play their role, we briefly go through the different transformation steps. Consider again that for the time quantifier differentiation (Proposition 1), for each time quantifier that occurs, the length of the subformula in the scope of this quantifier is doubled. As asserted in Section 8.3, in the most negative case, when their scopes are rather global and overlapping, the length of the formula can roughly become multiplied by 2^n . However, in the most positive case, when quantifiers have rather local and disjoint scopes, this multiplies the length of the formula only by a factor 2. This can be expressed by a lower bound of about $2c$ and an upper bound of about $2^n c$. When some measure of locality of scopes s would be possible, this could be rephrased as indication of the form $2c + s(2^n c - 2c) = (2 + s(2^n - 2))c = (s2^n + (1-s)2)c$, where $0 \leq s \leq 1$ with 0 means very local, disjoint scopes and 1 means maximally global, overlapping and nested scopes. Part of the added complexity is reduced (Proposition 2) by the elimination of past-future mixed time ordering atoms that are created. This reduction depends on the fraction f of state-related atoms among the atoms, with extremes 0 (only time-ordering atoms), and 1 (only state-related atoms). Including this factor to the formula may obtain $(2 + fs(2^n - 2))c$.

Transformation into prenex form (Proposition 3) and conjunctive normal form (Proposition 4) can depend, on the number of boolean connectives in the original formula (versus the number of quantifiers). If this number is low, then complexity can also be limited in this transformation. As stated before, transformation into implication form (Proposition 5) and concentrating quantifiers (Proposition 6) does not change complexity. The distribution of quantifiers within the implications (Proposition 7) adds complexity for half of them, the existential ones. The growth of the part of the formula addressed for an existential quantifier for a variable x (its scope) is about $\frac{1}{2} n(x) 2^{n(x)} = n(x) 2^{n(x)-1}$ with $n(x)$ the number of implications in which x occurs. Here $n(x)$ can vary from 1 for a very local scope to in proportion with the length of the conjunctive normal form formula for a global scope. As the latter already can be exponential in the length of the original formula, in the combined worst case, roughly spoken, this would imply double exponential for the length of the resulting formula compared to the length of the original formula (one for the conjunctive normal form transformation, and one for the distribution of quantifiers over the implications).

As for specific cases, the complexity shown in reality depends on a number of factors as indicated above, in realistic cases the resulting formula may be less complex than indicated by the worst case analysis in shown Section 8.3. As an example, a high number of quantifiers can lead to high complexity by the step to distribute quantifiers over the implications, and a high number of boolean connectives can lead to high complexity in the transformation into conjunctive normal form. However, within a formula of given length there is a trade-off in the sense that not both can be high, so in a complexity analysis this balance between the two has to be taken into account. In general, a combined worst case analysis may suffer from the same pessimism shown by applications of probability theory

where a (too) low probability is obtained by multiplication of probabilities under the assumption of independence, which may not be a correct assumption. An interesting challenge is to perform a realistic formal complexity analysis, taking into account factors and trade-offs as those indicated above. This challenge is left for future work.

There is a limitation of the format to which this transformation procedure can be applied. One of them is that calculations over time were excluded to avoid atoms of the form $u_i + 10 > v_j$ which are past-future mixed atoms, i.e., not past or future atoms. However, if the time frame is assumed to be discrete, such mixed atoms can be replaced by disjunctions of the form:

$$\bigvee_{a, b \geq 0 \text{ with } a + b = 10} [u_i + a = t-1 \wedge t + b + 1 > v_j]$$

which consists only of past atoms and future atoms.

Future work will include further investigation of the complexity of the procedure for realistic cases, incorporation of more efficiency improving simplification steps wherever possible, and a full implementation of the procedure.

References

- Ashby, W.R. (1952). *Design for a Brain*. Chapman & Hall, London. Revised edition 1960.
- Barringer, H., M. Fisher, D. Gabbay, R. Owens, and M. Reynolds (1996). *The Imperative Future: Principles of Executable Temporal Logic*. John Wiley & Sons, 1996.
- Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J., (2009). Specification and Verification of Dynamics in Agent Models. In: *International Journal of Cooperative Information Systems*, 2009, to appear. Shorter version in: *Proceedings of the Sixth International Conference on Intelligent Agent Technology, IAT'06*. IEEE Computer Society Press, 2006, pp. 247-254.
- Bosse, T., Jonker, C.M., and Treur, J., (2007). On the use of Organisation Modelling Techniques to Address Biological Organisation. *Multi-Agent and Grid Systems Journal*, vol. 3, 2007, pp. 199-223.
- Dixon, C., Bolotov, A., and Fisher, M. (2005). Alternating Automata and Temporal Logic Normal Forms. *Annals of Pure and Applied Logic*, vol. 135(1-3), 2005, pp. 263-285.
- Fisher, M. (1997). A Normal Form for Temporal Logics and its Applications in Theorem-Proving and Execution. *Journal of Logic and Computation*, vol. 7, 1997, pp. 429-456.
- Fisher, M. (2005). Temporal Development Methods for Agent-Based Systems, *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 10, pp. 41-66.
- Gabbay, D.M. (1989). The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems. In B. Banieqbal, H. Barringer, and A. Pnueli, (eds), *Proceedings of the 1st Conference on Temporal Logic in Specification*. Lecture Notes in Computer Science, vol. 398. Springer Verlag, 1989, pp. 409-448.
- Gabbay, D.M., Hodkinson, I., Reynolds, M. (1994). *Temporal Logic: Mathematical and Computational Aspects*, Vol. 1. Clarendon Press, Oxford, 1994.
- Galton, A. (2003). Temporal Logic. *Stanford Encyclopedia of Philosophy*, URL: <http://plato.stanford.edu/entries/logic-temporal/#2>
- Galton, A. (2006). Operators vs Arguments: The Ins and Outs of Reification. *Synthese*, vol. 150, (2006), pp. 415-441.

- Gelder, T.J. van, and Port, R.F., (1995). It's About Time: An Overview of the Dynamical Approach to Cognition. In: Port and van Gelder, 1995, pp. 1-43.
- Herlea Damian, D.E., Jonker, C.M., Treur, J., and Wijngaards, N.J.E., (2005). Integration of Behavioural Requirements Specification within Compositional Knowledge Engineering. *Knowledge-Based Systems Journal*, vol. 18, 2005, pp. 353-365.
- Hodkinson, I., and Reynolds, M. (2005). Separation - Past, Present and Future. In: *We Will Show Them: Essays in Honour of Dov Gabbay, Vol 2*. S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, and J. Woods (eds.), pp. 117-142, College Publications, 2005.
- Jonker, C.M., and Treur, J. (2002). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *International Journal of Cooperative Information Systems*, vol. 11, 2002, pp. 51-92. Preliminary, shorter version in: Roever, W.P. de, Langmaack, H., Pnueli, A. (eds.), (1998). *Proceedings of the International Workshop on Compositionality, COMPOS'97*. Lecture Notes in Computer Science, vol. 1536, Springer Verlag, 1998, pp. 350-380.
- Kelso, J.A.S. (1995). *Dynamic Patterns: the Self-Organisation of Brain and Behaviour*. MIT Press, Cambridge, Mass.
- Kowalski, R. A. and F. Sadri (1997). Reconciling the Event Calculus and the Situation Calculus, *Journal of Logic Programming* 31, 39–58.
- Kowalski, R. A. and M. J. Sergot: (1986). A Logic-based Calculus of Events. *New Generation Computing* 4, 67–95.
- Levesque, H., F. Pirri, and R. Reiter (1998). Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence*, 2(3-4):159-178.
- McCarthy, J. and P. J. Hayes (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: B. Melzer and D. Michie (eds.), *Machine Intelligence 4*, Edinburgh University Press, Edinburgh, pp. 463-502.
- Port, R.F., Gelder, T. van (eds.) (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, Mass.
- Reichgelt, H. (1987). Semantics for Reified Temporal Logic. In: J. Hallam and C. Mellish (eds.), *Advances in Artificial Intelligence*. Wiley, pp. 49–61.
- Reichgelt, H.: (1989), A Comparison of First Order and Modal Logics of Time. In: P. Jackson, H. Reichgelt, and F. van Harmelen (eds.), *Logic-Based Knowledge Representation*. MIT Press, Cambridge, MA, Chap. 7, pp. 143–176.
- Reiter, R. (2001). *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.
- Roever, W.P. de, Langmaack, H., Pnueli, A. (eds.), (1998). *Proceedings of the International Workshop on Compositionality, COMPOS'97*. Lecture Notes in Computer Science, vol. 1536, Springer Verlag, 1998.
- Sharpanskykh, A., and Treur, J., Verifying Interlevel Relations within Multi-Agent Systems. In: Brewka, G., Coradeschi, S., Perini, A., and Traverso, P. (eds.), *Proc. of the 17th European Conference on Artificial Intelligence, ECAI'06*. IOS Press, 2006, pp. 290-294.
- Vila, L., Reichgelt, H. (1996). The Token Reification Approach to Temporal Reasoning. *Artificial Intelligence*, vol. 83, pp. 59-74.

Appendix A Proofs

The proofs of most propositions are rather straightforward from the detailed information stated in the propositions. Only the proofs for Section 7 require a bit more work.

Definition (Accumulating)

The family of implications $A_i \rightarrow B_i$, $i \in I$ is called *accumulating* when for all $i, j \in I$ with $i \neq j$ it holds that there exists a k such that $A_i \wedge A_j \models A_k$ and $B_k \models B_i \wedge B_j$. An equivalent formulation is: for every finite nonempty subset $J \subseteq I$ there exists a k such that $\bigwedge_{j \in J} A_j \models A_k$ and $B_k \models \bigwedge_{j \in J} B_j$.

Lemma 1

If the family of implications $A_i \rightarrow B_i$, $i \in I$ is accumulating, and the free variable x does not occur in the A_i , then

$$\exists x \bigwedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{i \in I} \exists x [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{i \in I} [A_i \rightarrow \exists x B_i]$$

Proof

Semantic proof.

The second \Leftrightarrow is simple, and also the first \Leftrightarrow from left to right.

We concentrate on the first \Leftrightarrow from right to left.

Suppose M is a model and

$$M \models \bigwedge_{i \in I} \exists x [A_i \rightarrow B_i]$$

We have to prove

$$M \models \exists x \bigwedge_{i \in I} [A_i \rightarrow B_i]$$

Let $J \subseteq I$ be the following subset

$$J = \{ i \in I \mid M \models A_i \}$$

If J is nonempty, then a $k \in I$ exists such that $\bigwedge_{j \in J} A_j \models A_k$ and $B_k \models \bigwedge_{j \in J} B_j$. From $\bigwedge_{j \in J} A_j \models A_k$ it follows that $M \models A_k$ (i.e., $k \in J$). From

$$M \models \exists x [A_k \rightarrow B_k]$$

it follows that an assignment a to x in M exists such that $M[a/x] \models A_k \rightarrow B_k$. But since $M \models A_k$, then

$$M[a/x] \models B_k$$

Therefore, by $B_k \models \bigwedge_{j \in J} B_j$ it follows

$$M[a/x] \models \bigwedge_{j \in J} B_j$$

and hence also

$$M[a/x] \models \bigwedge_{j \in J} [A_j \rightarrow B_j]$$

For all $j \in I \setminus J$ it holds $M \models \neg A_j$. Therefore the implications $A_j \rightarrow B_j$ with $j \in I \setminus J$ are true in M for any assignment to x , so also for a :

$$M[a/x] \models \bigwedge_{j \in I \setminus J} [A_j \rightarrow B_j]$$

If J is empty this proves what has to be proven. If J is nonempty, combining the latter with with

$$M[a/x] \models \bigwedge_{j \in J} [A_j \rightarrow B_j]$$

as found above, this implies

$$M[a/x] \models \bigwedge_{j \in I} [A_j \rightarrow B_j]$$

so that

$$M \models \exists x \bigwedge_{j \in I} [A_j \rightarrow B_j] \quad \blacksquare$$

Lemma 2

a) For any family of implications $A_i \rightarrow B_i$, $i \in I$ there exist an accumulating family of formulae $A'_j \rightarrow B'_j$, $j \in J$ such that every A'_j is composed of A_i and every B'_j is composed of B_i and the following equivalence holds:

$$\bigwedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{j \in J} [A'_j \rightarrow B'_j]$$

b) More specifically, the index set J and the family $A'_j \rightarrow B'_j, j \in J$ in a) can be taken as follows:

$$\begin{aligned} J &= \mathcal{P}(I) && \text{(the power set of } I \text{)} \\ A'_j &= \bigwedge_{i \in j} A_i \\ B'_j &= \bigwedge_{i \in j} B_i \end{aligned}$$

Proof

Take the family $A'_j \rightarrow B'_j, j \in J$ as specified in b). For given $j_1, j_2 \in J$ take $k = j_1 \cup j_2$. Then

$$A'_{j_1} \wedge A'_{j_2} \Leftrightarrow \bigwedge_{i \in j_1} A_i \wedge \bigwedge_{i \in j_2} A_i \Leftrightarrow \bigwedge_{i \in j_1 \cup j_2} A_i = A'_k$$

and similarly

$$B'_{j_1} \wedge B'_{j_2} \Leftrightarrow B'_k$$

Therefore the family is accumulating.

Next from the equivalence to be proven

$$\bigwedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \bigwedge_{j \in J} [A'_j \rightarrow B'_j]$$

The direction \Leftarrow is simple as for each $i \in I$ the singleton $\{i\}$ is element of J , and

$$\begin{aligned} A'_{\{i\}} &= A_i \\ B'_{\{i\}} &= B_i \end{aligned}$$

For the direction \Rightarrow of the equivalence, this follows from (for any $j \in J$):

$$\bigwedge_{i \in j} [A_i \rightarrow B_i] \Rightarrow [\bigwedge_{i \in j} A_i \rightarrow \bigwedge_{i \in j} B_i] \Leftrightarrow A'_j \rightarrow B'_j \quad \blacksquare$$

Proposition 8 (Distributing Quantifiers over Conjunctions of Implications)

Let φ be a formula in the form of a conjunction of implications

$$\bigwedge_{i \in I} [A_i \rightarrow B_i]$$

and let x be a variable occurring in φ . Then the following hold.

a) The formula φ is equivalent to a formula φ' of the form

$$\bigwedge_{j \in J} A'_j \rightarrow B'_j$$

such that if x occurs in one or more of the B_i it holds

$$\exists x \bigwedge_{j \in J} [A'_j \rightarrow B'_j] \Leftrightarrow \bigwedge_{j \in J} \exists x [A'_j \rightarrow B'_j] \Leftrightarrow \bigwedge_{j \in J} [A'_j \rightarrow \exists x B'_j]$$

Moreover, every A'_j is a conjunction of A_i and every B'_j is a conjunction of B_i .

b) The formula φ is equivalent to a formula φ'' of the form

$$\bigwedge_{j \in J} A''_j \rightarrow B''_j$$

such that if the variable x occurs in one or more of the A_i it holds

$$\exists x \bigwedge_{j \in J} [A''_j \rightarrow B''_j] \Leftrightarrow \bigwedge_{j \in J} \exists x [A''_j \rightarrow B''_j] \Leftrightarrow \bigwedge_{j \in J} [\forall x A''_j \rightarrow B''_j]$$

Moreover, every A''_j is a disjunction of A_i and every B''_j is a disjunction of B_i .

c) The formulae φ' and φ'' in a) and b) respectively can be obtained from φ as follows.

$$\begin{aligned} \bigwedge_{i \in I} [A_i \rightarrow B_i] &\rightarrow \bigwedge_{j \in \mathcal{P}(I)} [\bigwedge_{i \in j} A_i \rightarrow \bigwedge_{i \in j} B_i] \\ \bigwedge_{i \in I} [A_i \rightarrow B_i] &\rightarrow \bigwedge_{j \in \mathcal{P}(I)} [\bigvee_{i \in j} A_i \rightarrow \bigvee_{i \in j} B_i] \end{aligned}$$

The following transformation rules handle existential quantifiers for variables in one or more of the B_i , respectively in one or more of the A_i . Here \mathcal{P} denotes taking the power set.

(1) if x occurs in the B_i but does not occur in the A_i :

$$\exists x \bigwedge_{i \in I} [A_i \rightarrow B_i] \rightarrow \bigwedge_{j \in \mathcal{P}(I)} \exists x [\bigwedge_{i \in j} A_i \rightarrow \bigwedge_{i \in j} B_i]$$

$$\exists x [\wedge_{i \in J} A_i \rightarrow \wedge_{i \in J} B_i] \quad \rightarrow \quad [\wedge_{i \in J} A_i \rightarrow \exists x \wedge_{i \in J} B_i]$$

(2) if x occurs in the A_i but does not occur in the B_i :

$$\exists x \wedge_{i \in I} [A_i \rightarrow B_i] \quad \rightarrow \quad \wedge_{j \in P(I)} \exists x [\vee_{i \in j} A_i \rightarrow \vee_{i \in j} B_i]$$

$$\exists x [\vee_{i \in J} A_i \rightarrow \vee_{i \in J} B_i] \quad \rightarrow \quad [\forall x [\vee_{i \in J} A_i] \rightarrow \vee_{i \in J} B_i]$$

d) The following transformation rules handle universal quantifiers for variables in one or more of the B_i , respectively in one or more of the A_i :

(1) if x occurs in the A_i or in the B_i :

$$\forall x \wedge_{i \in I} [A_i \rightarrow B_i] \quad \rightarrow \quad \wedge_{i \in I} \forall x [A_i \rightarrow B_i]$$

(2) if x occurs in the B_i but does not occur in the A_i :

$$\forall x [A_i \rightarrow B_i] \quad \rightarrow \quad A_i \rightarrow \forall x B_i$$

(3) if x occurs in the A_i but does not occur in the B_i :

$$\forall x [A_i \rightarrow B_i] \quad \rightarrow \quad [\exists x A_i] \rightarrow B_i$$

Proof

Here it is shown how they are applied to handle the existential quantifiers.

(a) For the case of a future time variable v_k occurring (only) in one or more of the consequents, the Lemmas can be applied in a straightforward manner. The transformation rules are based on

$$\begin{aligned} \exists v_k < t \wedge_{i \in I} [A_i \rightarrow B_i] &\Leftrightarrow \wedge_{i \in J} \exists v_k < t [\wedge_{i \in j} A_i \rightarrow \wedge_{i \in j} B_i] \\ &\Leftrightarrow \wedge_{i \in J} [\wedge_{i \in j} A_i \rightarrow \exists v_k < t \wedge_{i \in j} B_i] \end{aligned}$$

(b) For the case of a past time variable u_k , occurring (only) in one or more of the antecedents, the following transformation can be made:

$$\wedge_{i \in I} [A_i \rightarrow B_i] \quad \Leftrightarrow \wedge_{i \in I} [\neg B_i \rightarrow \neg A_i]$$

Now, applying Lemma 2 to the family $\neg B_i \rightarrow \neg A_i$, $i \in I$ provides an accumulating family $A'_j \rightarrow B'_j$, $j \in J$ such that every A'_j is composed of $\neg A_i$ and every B'_j is composed of $\neg B_i$ such that:

$$\wedge_{i \in I} [\neg B_i \rightarrow \neg A_i] \quad \Leftrightarrow \wedge_{j \in J} [B'_j \rightarrow A'_j]$$

Applying Lemma 1 yields

$$\begin{aligned} \exists u_k < t \wedge_{i \in I} [A_i \rightarrow B_i] &\Leftrightarrow \exists u_k < t \wedge_{i \in I} [\neg B_i \rightarrow \neg A_i] \\ &\Leftrightarrow \exists u_k < t \wedge_{j \in J} [B'_j \rightarrow A'_j] \\ &\Leftrightarrow \wedge_{i \in J} \exists u_k < t [B'_j \rightarrow A'_j] \\ &\Leftrightarrow \wedge_{i \in J} [B'_j \rightarrow \exists u_k < t A'_j] \\ &\Leftrightarrow \wedge_{i \in J} [\neg \exists u_k < t A'_j \rightarrow \neg B'_j] \\ &\Leftrightarrow \wedge_{i \in J} [\forall u_k < t \neg A'_j \rightarrow \neg B'_j] \end{aligned}$$

Here it can be taken

$$A'_j \quad \Leftrightarrow \wedge_{i \in j} \neg A_i$$

$$B'_j \quad \Leftrightarrow \wedge_{i \in j} \neg B_i$$

so

$$\neg A'_j \quad \Leftrightarrow \neg \wedge_{i \in j} \neg A_i \quad \Leftrightarrow \vee_{i \in j} A_i$$

$$\neg B'_j \quad \Leftrightarrow \neg \wedge_{i \in j} \neg B_i \quad \Leftrightarrow \vee_{i \in j} B_i$$

Therefore a transformation rule can be based on

$$\begin{aligned} \exists u_k < t \wedge_{i \in I} [A_i \rightarrow B_i] &\Leftrightarrow \wedge_{i \in J} \exists u_k < t [\vee_{i \in j} A_i \rightarrow \vee_{i \in j} B_i] \\ &\Leftrightarrow \wedge_{i \in J} [\forall u_k < t \vee_{i \in j} A_i \rightarrow \vee_{i \in j} B_i] \end{aligned} \quad \blacksquare$$

Appendix B Alternative Approach for Quantifier Distribution

This Appendix provides an alternative approach for the quantifier distribution addressed in Section 7.

Definition (Mutually Exclusive)

The family of formulae $A_i, i \in I$ is called *mutually exclusive* when for all $i, j \in I$ with $i \neq j$ it holds that $A_i \wedge A_j$ is invalid, or, in other words, $\neg(A_i \wedge A_j)$ is a validity, or $A_i \models \neg A_j$.

Lemma 3

If the family of formulae $A_i, i \in I$ is mutually exclusive, and the free variable x does not occur in them, then

$$\exists x \wedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \wedge_{i \in I} \exists x [A_i \rightarrow B_i] \Leftrightarrow \wedge_{i \in I} [A_i \rightarrow \exists x B_i]$$

Proof

Semantic proof.

The second \Leftrightarrow is simple, and also the first \Leftrightarrow from left to right.

We concentrate on the first \Leftrightarrow from right to left.

Suppose M is a model and

$$M \models \wedge_{i \in I} \exists x [A_i \rightarrow B_i]$$

We have to prove

$$M \models \exists x \wedge_{i \in I} [A_i \rightarrow B_i]$$

If for all of the A_i it holds $M \models \neg A_i$ then for any x these implications are true in M .

Now suppose for one A_k it holds $M \models A_k$. Then by mutual exclusiveness for all $j \neq k$ it holds $M \models \neg A_j$. Therefore the implications $A_j \rightarrow B_j$ with $j \neq k$ are true in M for any assignment to x . Now from

$$M \models \wedge_{i \in I} \exists x [A_i \rightarrow B_i]$$

It follows that an assignment a to x in M exists such that $M[a/x] \models A_k \Rightarrow B_k$. But then

$$M[a/x] \models \wedge_{i \in I} A_i \rightarrow B_i$$

Therefore

$$M \models \exists x \wedge_{i \in I} [A_i \rightarrow B_i] \quad \blacksquare$$

Lemma 4

For any family of formulae $A_i, B_i, i \in I$ there exist a family of formulae $A'_j, B'_j, j \in J$ with $A'_j, j \in J$ mutually exclusive such that every A'_j is composed of A_i and every B'_j is composed of B_i and the following equivalence holds:

$$\wedge_{i \in I} [A_i \rightarrow B_i] \Leftrightarrow \wedge_{j \in J} [A'_j \rightarrow B'_j]$$

More specifically, the index set J and the family $A'_j, B'_j, j \in J$ can be taken as follows:

$$J = \mathcal{P}(I) \quad (\text{the power set of } I)$$

$$A'_j = \wedge_{i \in j} A_i \wedge \wedge_{i \in I \setminus j} \neg A_i$$

$$B'_j = \wedge_{i \in j} B_i$$

Proof

First it is shown why the $A'_j, j \in J$ are mutually exclusive. Let $j, k \in J$ be given with $j \neq k$. Then there exist an $i \in I$ in one and not in the other, say such that $i \in j$ but $i \notin k$. Then A_i occurs as a conjunct in A'_j and $\neg A_i$ occurs as a conjunct in A'_k . Therefore $A'_j \wedge A'_k$ has both A_i and $\neg A_i$ as a conjunct and thus is invalid. Now the equivalence is addressed.

\Rightarrow

Suppose $j \in J$ is given, then for all $i \in j$ we have

$$A_i \rightarrow B_i$$

Therefore

$$\bigwedge_{i \in J} A_i \rightarrow \bigwedge_{i \in J} B_i$$

and certainly, as its antecedent is stronger

$$\bigwedge_{i \in J} A_i \wedge \bigwedge_{i \in I \setminus J} \neg A_i \rightarrow \bigwedge_{i \in J} B_i$$

This proves \Rightarrow .

\Leftarrow

This follows from a kind of resolution process. From

$$\bigwedge_{i \in J} [A_i \rightarrow B_i]$$

By downward induction on the size n of the union J of any two disjoint subsets J_1 and J_2 of I it is proven that

for all $J_1, J_2 \in J$ with $J_1 \cap J_2 = \emptyset$ and $\#(J_1 \cup J_2) = n$ it holds

$$\bigwedge_{i \in J_1} A_i \wedge \bigwedge_{i \in J_2} \neg A_i \rightarrow \bigwedge_{i \in J} B_i$$

Then the result follows for sets J_1 of size 1 and J_2 of size 0.

For starting point: $\#(J) = \#(I)$ this is exactly the premisses we have.

Now suppose the induction hypothesis holds for n .

Let any two disjoint sets $J_1, J_2 \in J$ be given with $\#(J_1 \cup J_2) = n-1$. Take any $k \in I$ outside $J_1 \cup J_2$ and put $J_1' = J_1 \cup \{k\}$, $J_2' = J_2 \cup \{k\}$. Then both J_1', J_2' and J_1, J_2' are pairs of sets with union of size n , so the induction hypothesis can be applied on both pairs:

$$\bigwedge_{i \in J_1'} A_i \wedge \bigwedge_{i \in J_2} \neg A_i \rightarrow \bigwedge_{i \in J_1'} B_i$$

$$\bigwedge_{i \in J_1} A_i \wedge \bigwedge_{i \in J_2'} \neg A_i \rightarrow \bigwedge_{i \in J_1} B_i$$

This is equivalent to

$$\bigwedge_{i \in J_1} A_i \wedge A_k \wedge \bigwedge_{i \in J_2} \neg A_i \rightarrow \bigwedge_{i \in J_1} B_i \wedge B_k$$

$$\bigwedge_{i \in J_1} A_i \wedge \neg A_k \wedge \bigwedge_{i \in J_2} \neg A_i \rightarrow \bigwedge_{i \in J_1} B_i$$

From this it follows by a form of resolution that

$$\bigwedge_{i \in J_1} A_i \wedge \bigwedge_{i \in J_2} \neg A_i \rightarrow \bigwedge_{i \in J_1} B_i$$

This proves the induction step.

By induction the general statement is obtained, and for $n = 1$ the result is found that it holds

$$\bigwedge_{i \in I} [A_i \rightarrow B_i]$$

■