Compositional Verification of Knowledge-Based Systems in Temporal Epistemic Logic

Joeri Engelfriet, Catholijn M. Jonker, Jan Treur

Vrije Universiteit Amsterdam
Faculty of Mathematics and Computer Science, Artificial Intelligence Group
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
URL: http://www.cs.vu.nl/~{joeri,jonker,treur}, Email: {joeri,jonker,treur}@cs.vu.nl

1 Introduction

It is a recent trend in the literature on verification to study the use of compositionality and abstraction to structure the process of verification; for example, see (Abadi and Lamport, 1993; Hooman, 1994; Dams, Gerth and Kelb, 1996). In (Cornelissen, Jonker and Treur, 1997) a compositional verification method was introduced for logic-based specifications of knowledge-based systems. The current paper discusses the requirements for the choice and use of a suitable logic with which verification proofs of such compositional reasoning systems can be formalized. For the particular application of the logic the following requirements for the logic itself and for the use of the logic are of importance:

- compositional structure: proofs can be structured in a compositional manner, in accordance with the compositional structure of the system design.
- dynamics and time: dynamic properties can be expressed, reasoning and induction over time is possible.
- incomplete information states can be expressed.
- transparency: the proof system and the semantics are transparent and not unnecessarily complicated.

In the following sections, temporal epistemic logic (TEL) is shown to be a suitable logic; this logic was introduced in (Engelfriet and Treur 1996a, 1996b); see also (Engelfriet, 1996) and (Engelfriet and Treur, 1997). By choosing temporal epistemic logic, a choice was also made for a discrete and linear time structure and for time to be global, which is a suitable choice for the formalization of verification proofs of sequential reasoning systems.

The structure of the paper is as follows. In Section 2 the compositional verification method for knowledge-based systems is briefly described and an example is given. In Section 3 the temporal epistemic logic is defined. Section 4 discusses compositional temporal theories, Section 5 compositional proof structures, and Section 6 focuses on how to treat non-classical semantics related to default persistence.

2 Compositional Verification

The purpose of verification is to prove that, under a certain set of assumptions, a system satisfies a certain set of properties, for example the design requirements. In the approach introduced in (Cornelissen et al., 1997), this is done by mathematical proof (i.e., a proof in the form mathematicians are accustomed to do), which proves that the specification of the system together with the assumptions implies the properties that the system needs to fulfill. A compositional reasoning system (for example designed using the modelling framework DESIRE, see Brazier, Treur, Wijngaards and Willems, 1995, 1996) can be viewed and specified at different levels of abstraction, see Figure 1. Viewed from the top level, denoted by L_0, the complete system is one component, in this case diagnostic_reasoning, with interfaces, whereas internal information and processes are left unspecified at this level of abstraction (information and process hiding). At the next

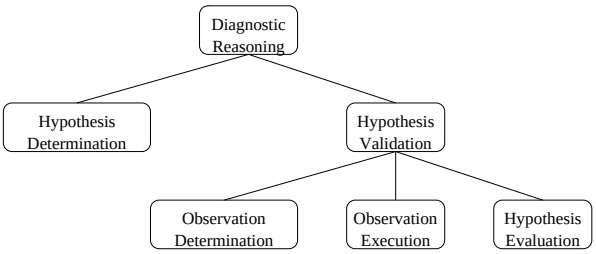

Figure 1 Levels of abstraction in the diagnostic reasoning system

level of abstraction, L_1, the internal structure of diagnostic_reasoning is given, but the details of the subcomponents are hidden; see Figure 2, upper part. At the next lower level of abstraction, L_2, the component hypothesis_validation is specified as a composition of the sub-components, information links, and task control, depicted in Figure 2, lower part.

The compositional verification method takes into account this compositional structure during the verification process. Verification of a composed component is done using properties of the sub-components it embeds and the component's specification (which specifies how it is composed of its sub-components). The assumptions on its subcomponents under which the component functions properly, are properties to be proven for these sub-components. This implies that properties at different levels of abstraction are involved in the verification process. These properties have hierarchical logical relations in the sense that at each level, given the component's specification, a property is logically implied by (a conjunction of) the lower level properties that relate to it in the hierarchy (see Figure 3).

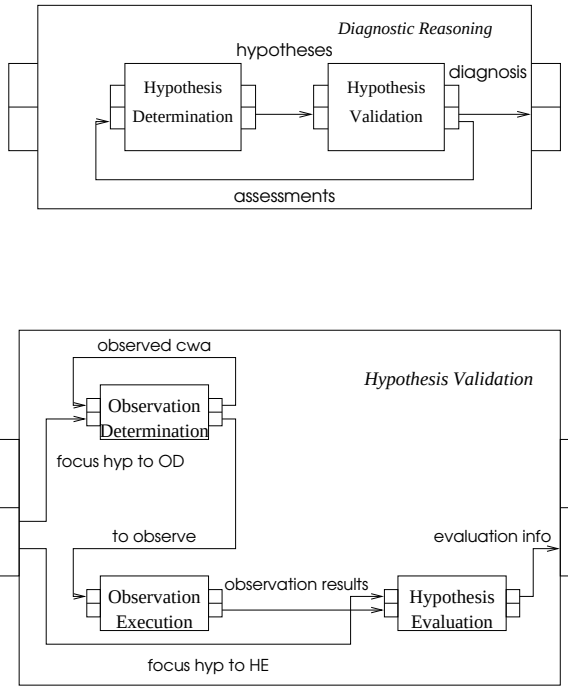

Figure 2 Composition at two levels of abstraction

Consider, for example, the property system assessment conservative, which means that the diagnostic assessments made by the diagnostic reasoning system do not change over time, e.g., a hypothesis rejected by the system stays rejected. This property depends directly on the same property of the sub-component hypothesis_validation. This component validates a hypothesis which was focused upon by the component hypothesis_determination. In turn, this property depends on the same property of its sub-component hypothesis_evaluation, first ignoring the other sub-components of hypothesis_validation that have to do with observation determination and execution. Finally, conservativity depends on the property that the results of the observations made by the primitive component observation_execution do not change over time. For the definitions of the properties in Figure 3, see (Cornelissen et al., 1997). Induction over time is needed to prove the system's effectiveness and termination. Effectiveness means that the system will always give a diagnosis if a diagnosis is possible.

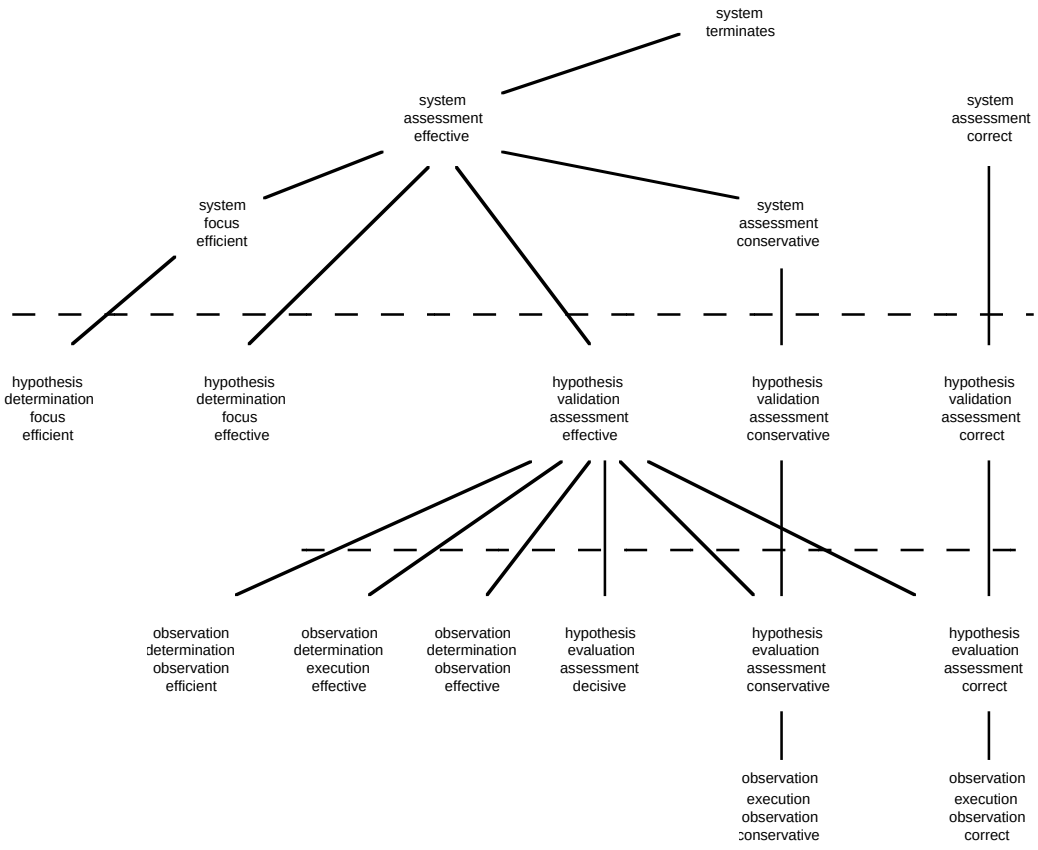

Figure 3 Logical relations between properties at different levels of abstraction for the diagnostic reasoning model

The *compositional verification method* can be formulated in more detail as follows:

A. Verifying one abstraction level against the other
For each abstraction level the following procedure is followed:
1. Determine which properties are of interest (for the higher level).
2. Determine assumptions (at the lower level) that guarantee these properties.
3. Prove the properties on the basis of these assumptions.

B. Verifying a primitive component
For primitive knowledge-based components a number of verification techniques exist in the literature, see for example (Treur and Willems, 1994).

C. The overall verification process
To verify the complete system:
1. Determine the properties are that are desired for the whole system.
2. Apply the above procedure **A** iteratively until primitive components are reached.
   In the iteration the desired properties of abstraction level L_i are either:
   - those determined in step **A1**, if i = 0, or
   - the assumptions made for the higher level L_{i-1}, if i > 0
3. Verify the primitive components according to **B**.

The results of verification are:

- Properties and assumptions at the different abstraction levels.
- Logical relations between the properties of different abstraction levels (cf. Figure 3).

Note that both static and dynamic properties and connections between them are covered. Furthermore, process and information hiding limits the complexity of the verification per abstraction level.

3 Temporal Epistemic Logic

In this section we introduce a logic that can be used to formalize the dynamic aspects of reasoning and the incomplete information states that play a role: temporal epistemic logic. Our approach is in line with what in (Finger and Gabbay, 1992) is called temporalising a given logic; in our case the given logic is S5-logic. As the base language in which the reasoning system can express its knowledge and conclusions, we will take a propositional language. The following definition formalizes information states and a temporalization of these states, using linear discrete time with a starting point. For convenience we will take the set of natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ as the time frame.

Definition 3.1 (temporal epistemic model)
a) A *signature* Σ is an ordered sequence of (propositional) atom names. An *epistemic state*, or *information state*, based on Σ, is a set of propositional models of signature Σ.
The set of information states based on Σ is denoted by $IS(\Sigma)$, or shortly IS.
b) Let Σ be a signature. A (propositional) *temporal epistemic model* M of signature Σ is a mapping $M: \mathbb{N} \rightarrow IS(\Sigma)$. We will sometimes use the notation $(M_t)_{t \in \mathbb{N}}$ for M.

Propositional formulae can be evaluated in epistemic states at any point in time: a propositional formula α is (known to be) true in an epistemic state M, denoted $M \models \alpha$, if $m \models \alpha$ for all $m \in M$. We will need a language to express changes over time. To this end in (Engelfriet and Treur, 1996a, 1996b) a temporal epistemic language and its semantics were introduced. The temporal operators C, X, Y, F and G are used. Intuitively, the temporal formula $F\alpha$ is true at time t means that viewed from time point t, the formula α will be true (or *known*) at *some* time in the future (in *some* future information state), $G\alpha$ is true at time t means that viewed from time point t, the formula α will be true (or *known*) at *all* time points in the future, and $X\alpha$ is true at time t means that α will be true in the next information state. The operator Y means "true at the previous time point". Furthermore, an operator is required that expresses the fact that *currently* α is true (in the *current* information state); this is the operator C. The C operator is very similar to the modal K operator, so for instance the formula $\neg C\alpha \wedge \neg C\neg \alpha$ denotes that α is unknown (i.e., neither known to be true nor known to be false). In contrast with traditional temporal logics, it is not the case that $G\alpha \equiv \neg F\neg \alpha$ (take a temporal model in which α is always unknown). For more details, see (Engelfriet and Treur, 1996a, 1996b, 1997, Engelfriet 1996). For temporal epistemic logic different entailment relations can be used, both classical and non-classical; see e.g., (Engelfriet and Treur, 1997; Engelfriet, 1996).

Example 3.2 (a conservativity property)
The property system assessment conservative (see Figure 3) can be expressed by the following temporal epistemic formulae for $h \in HYPS$:

$$Y(\text{output\_DR\_confirmed}(h)) \rightarrow C(\text{output\_DR\_confirmed}(h))$$
$$Y(\text{output\_DR\_rejected}(h)) \rightarrow C(\text{output\_DR\_rejected}(h))$$

These formulae express that once a conclusion about a rejected or confirmed hypothesis has occurred at the system's output interface, this conclusion will persist. The conjunction of these temporal epistemic logic formulae is named by **system_assessment_conservative**. Although this language is not propositional, one can consider an expression like **output_DR_confirmed(h)**, for some **h ∈ HYPS**, as a notation for a propositional atom **output_DR_confirmed_h**. This makes the language propositional.

4 Compositional temporal theories

In order to embed the compositional verification proofs in temporal epistemic logic, a reasoning system specification is translated into a temporal theory. As a requirement on this translation we impose that the compositional structure is preserved. This means that instead of one global temporal theory, each component in the hierarchy is translated into a separate temporal theory for this component. Therefore, we introduce collections of sub-languages and collections of temporal theories that are labelled by the set of components **COMP**. A language for a component defines the terms in which information in its input and output interface can be expressed.

Definition 4.1 (language composition)
Let **COMP** be a set of component names with a binary sub-component relation **sub**. *Primitive* components are elements **D ∈ COMP** for which no **C ∈ COMP** exists with **C sub D**. The other components are called *composed*.

A *language composition* is a collection of sub-languages $(L_C)_{C \in COMP}$. For convenience these languages are assumed disjoint.

The collection of *bridge languages* for the language composition $(L_C)_{C \in COMP}$ is the collection $(L^+_C)_{C \in COMP}$ defined for any component **D** by

$$L^+_D = L_D \cup \bigcup_{C \text{ sub } D} L_C$$

The *cumulative language composition* for the language composition $(L_C)_{C \in COMP}$ is the collection $(L^*_C)_{C \in COMP}$ defined for any component **D** by

$L^*_D = L_D \cup \bigcup_{C \text{ sub } D} L^*_C$ if **D** is a composed component

$L^*_D = L_D$ if **D** is a primitive component

Example 4.2 (language composition of the diagnostic reasoning system)
The languages for the different components of the diagnostic reasoning system are defined by (here **h ∈ HYPS**, **o ∈ OBS**):

L_{DR}	**output_DR_rejected(h), output_DR_confirmed(h)**
L_{HD}	**input_HD_rejected(h), input_HD_confirmed(h), output_HD_focus(h)**
L_{HV}	**input_HV_focus(h), output_HV_rejected(h), output_HV_confirmed(h)**
L_{OD}	**input_OD_focus(h), input_OD_observed(o), output_OD_to_observe(o)**
L_{OE}	**input_OE_target(o), output_OE_o**
L_{HE}	**input_HE_o, output_HE_h**

These languages formalize the input and output interfaces of the components at different levels of abstraction. Note that for the top-level diagnostic_reasoning the information about focus hypotheses (that does occur in hypothesis_determination and hypothesis_validation) is hidden. For diagnostic_reasoning and the level of hypothesis_validation (as well as hypothesis_determination) the information about observations (that occurs in observation_determination, observation_execution and hypothesis_evaluation) is hidden.

Definition 4.3 (theory composition)
Let $(L_C)_{C \in COMP}$ be a language composition. A *compositional temporal theory* is a collection $(T_C)_{C \in COMP}$ where each temporal theory T_C is a theory in the language L^+_C.
Let $(T_C)_{C \in COMP}$ be a compositional temporal theory. The *collection of cumulative theories* $(T^*_C)_{C \in COMP}$ is defined for any component **D** as:

$T^*_D = T_D \cup \bigcup_{C \text{ sub } D} T^*_C$ if **D** is a composed component
$T^*_D = T_D$ if **D** is a primitive component

Example 4.4 (part of a theory composition; a composed component)
For each of the components of the diagnostic reasoning system its specification can be translated into a temporal theory. For example, as part of the theory T_{DR}, the information link diagnosis, which transfers the assessments made by the component hypothesis_validation (i.e., the truth values **true** or **false** of the output atoms) to the output of diagnostic_reasoning, can be formalized by the following formulae (where $h \in HYPS$):

$$C(DR\_uptodate(diagnosis)) \wedge Y(output\_HV\_confirmed(h)) \rightarrow C(output\_DR\_confirmed(h))$$

$$C(DR\_uptodate(diagnosis)) \wedge Y(output\_HV\_rejected(h)) \rightarrow C(output\_DR\_rejected(h))$$

$$C(DR\_uptodate(diagnosis)) \wedge Y(\neg\, output\_HV\_confirmed(h)) \rightarrow C(\neg\, output\_DR\_confirmed(h))$$

$$C(DR\_uptodate(diagnosis)) \wedge Y(\neg\, output\_HV\_rejected(h)) \rightarrow C(\neg\, output\_DR\_rejected(h))$$

These formulae express that if the link diagnosis is currently up to date, then the output of the component diagnostic_reasoning has the information of the output of hypothesis_validation at the previous time point.

Example 4.5 (part of a theory composition; a primitive component)
Primitive components can, for example, be specified by logical rules of the form 'conjunction of literals' implies 'literal'), as is the case in DESIRE. Consider the following rule of the knowledge base of the primitive component hypothesis_determination:

if not confirmed(h) **and** not rejected(h) **then** focus(h)

This rule can be formalized in **TEL** by (where $h \in HYPS$):

$$\phi \wedge Y(\neg\, input\_HD\_confirmed(h)) \wedge Y(\neg\, input\_HD\_rejected(h)) \rightarrow C(output\_HD\_focus(h))$$

where ϕ is a formula expressing control information that allows the rule to be used (for example, the component should be active).

5 Compositional proof structures

Verification proofs, such as those summarized in Figure 3, are composed of proofs at different levels of abstraction. These proofs involve properties of the components at these abstraction levels.

Definition 5.1 (collection of properties)
A *composition of properties* for a language composition $(L_C)_{C \in COMP}$ is a collection $(P_C)_{C \in COMP}$ where each P_C is a set of temporal statements in the language L_C.

Example 5.2 (a collection of conservativity properties)
The conservativity properties depicted in Figure 3 can be described by the following composition of properties (see also Example 3.2):

$\mathbf{P_{DR}}$ = { **system_assessment_conservative** }

$\mathbf{P_{HD}} = \varnothing$
$\mathbf{P_{HV}}$ = { **hypothesis_validation_assessment_conservative** }

$\mathbf{P_{OD}} = \varnothing$
$\mathbf{P_{OE}}$ = { **observation_execution_conservative** }
$\mathbf{P_{HE}}$ = { **hypothesis_evaluation_assessment_conservative** }

where the variants of assessment conservativity are defined as in Example 3.2, and the property **observation_execution_conservative** is defined as the conjunction of the following set of temporal formulae (with $\mathbf{o} \in$ **OBS** and $\mathbf{h} \in$ **HYPS**):

C(output_OE_o) $\rightarrow$ **X(output_OE_o)**
C(¬ output_OE_o) $\rightarrow$ **X(¬ output_OE_o)**

Definition 5.3 (compositional and global provability)
For the language composition $(\mathbf{L_C})_{\mathbf{C} \in \mathbf{COMP}}$, let a composition of properties $(\mathbf{P_C})_{\mathbf{C} \in \mathbf{COMP}}$ and a compositional temporal theory $(\mathbf{T_C})_{\mathbf{C} \in \mathbf{COMP}}$ be given. Let $\mathrel{|\!\sim}$ be an entailment relation for temporal epistemic logic.
a) The composition of properties $(\mathbf{P_C})_{\mathbf{C} \in \mathbf{COMP}}$ is *compositionally provable* with respect to $\mathrel{|\!\sim}$ from the compositional temporal theory $(\mathbf{T_C})_{\mathbf{C} \in \mathbf{COMP}}$ if for each component **D** the following holds:

$\mathbf{T_D} \cup \bigcup_{\mathbf{C\ sub\ D}} \mathbf{P_C} \mathrel{|\!\sim} \mathbf{P_D}$ if **D** is composed
$\mathbf{T_D} \mathrel{|\!\sim} \mathbf{P_D}$ if **D** is primitive

b) The composition of properties is *globally provable* with respect to $\mathrel{|\!\sim}$ from the compositional temporal theory $(\mathbf{T_C})_{\mathbf{C} \in \mathbf{COMP}}$ if for each component **D** the following holds:

$\mathbf{T^*_D} \mathrel{|\!\sim} \mathbf{P_D}$

For example, the collection of conservativity properties of Example 5.2 is globally provable with respect to $\mathrel{|\!\sim}$ from the compositional temporal theory $(\mathbf{T_C})_{\mathbf{C} \in \mathbf{COMP}}$.

Compositional provability does not necessarily imply global provability. However, the implication holds if the entailment relation satisfies, apart from reflexivity (if $\mathbf{V} \subseteq \mathbf{W}$, then $\mathbf{W} \mathrel{|\!\sim} \mathbf{V}$), the property of transitivity:

$\mathbf{T} \mathrel{|\!\sim} \mathbf{U}$ & $\mathbf{U} \mathrel{|\!\sim} \mathbf{W} \Rightarrow \mathbf{T} \mathrel{|\!\sim} \mathbf{W}$ (*Transitivity*)

for all sets of formulae **T**, **U**, **W**. It is well-known that transitivity and reflexivity imply monotonicity.

Proposition 5.4
If the entailment relation $\mathrel{|\!\sim}$ satisfies, in addition to reflexivity, transitivity, then compositional provability with respect to $\mathrel{|\!\sim}$ implies global provability with respect to $\mathrel{|\!\sim}$. In particular, if $\vdash$ is a classical provability relation for temporal epistemic logic, then compositional provability with respect to $\vdash$ implies global provability with respect to $\vdash$.

This proposition shows that for classical entailment the implication holds. But, for example, for the entailment relation taking into account minimal change the implication does not hold. In the light of these results, for compositional verification a classical proof system is the best choice.

6 Default persistence and revision

The conditions under which a classical inference relation can be used depend on the specific form of semantics. For example, in DESIRE a default persistence assumption has been made: it is only specified what has to be changed; all other information is meant to persist in time. An exception is made for information that has to be retracted because it was derived from information that does not hold anymore. In this section we discuss the manner in which default persistence and revision is treated within temporal epistemic logic.

In principle, a compositional specification can be formalized by executable temporal formulae. Roughly spoken executable temporal formulae are temporal formulae of the form

declarative past implies imperative future

For more details on this paradigm, and the different variants within, see (Barringer, Fisher, Gabbay and Hunter, 1993; Barringer, Fisher, Gabbay, Owens and Reynolds, 1996). For our purposes the following definition is chosen. *Simplified executable temporal formulae* are formulae of the form

past and present ⇒ present

The right hand side of these formulae **F** are called *heads*, denoted by **head(F)**; they are taken from the set

$$\begin{aligned} \mathbf{HEADS} \;=\; & \{\, \mathbf{CL} \mid \mathbf{L} \text{ propositional literal} \,\} \cup \\ & \{\, \neg\, \mathbf{CA} \wedge \neg\, \mathbf{C} \neg\, \mathbf{A} \mid \mathbf{A} \text{ propositional atom} \,\} \end{aligned}$$

The left hand side of **F** is called *body*, denoted by **body(F)**. Within the body, the 'past' part is a conjunction of temporal literals that are either of the form **YL**, **YYL**, **¬ YL** or **¬ YYL** with **L** a propositional literal. The 'present' part is a conjunction of temporal literals that are either of the form **CL** or **¬CL** with **L** a propositional literal.

The intended semantics of these formulae is that it is only specified what has to be changed. All other information is meant to persist (default persistence) in time, with an exception for information that has to be revised because it was derived from information that does not hold anymore. In principle this entails non-classical semantics. However, a translation is possible into temporal theories with classical semantics if a form of temporal completion (similar to Clark's completion) is applied:

Let **T** be a temporal theory consisting of simplified executable temporal formulae. For each **H ∈ HEADS** define

$$\mathbf{T_H} \;=\; \{\, \mathbf{F} \in \mathbf{T} \mid \mathbf{head(F)} = \mathbf{H} \,\}$$

Let **L** be a literal; define

$$\begin{aligned} \mathbf{tc(T_{CL})} \;=\; & [\, \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{CL}}\} \vee \\ & (\neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{C\sim L}}\} \wedge \\ & \neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{\neg CL \wedge \neg C\sim L}}\} \wedge \\ & \mathbf{YL}\ )\ ] \\ & \leftrightarrow \mathbf{CL} \end{aligned}$$

$$\begin{aligned} \mathbf{tc(T_{\neg CL \wedge \neg C\sim L})} \;=\; & [\, \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{\neg CL \wedge \neg C\sim L}}\} \vee \\ & (\neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{C\sim L}}\} \wedge \\ & \neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{CL}}\} \wedge \\ & \neg \mathbf{YL} \wedge \neg \mathbf{Y\sim L}\ )\ ] \\ & \leftrightarrow \neg \mathbf{CL} \wedge \neg \mathbf{C\sim L} \end{aligned}$$

Here **~L** denotes the complementary literal of **L**.

The *temporal completion* of **T** is defined by

$$\mathbf{tc(T) = \{\ tc(T_{CL})\ |\ L\ literal\ \} \cup \{\ tc(T_{\neg CL \wedge \neg C \sim L})\ |\ L\ literal\ \}}$$

Under a consistency assumption the right part $\mathbf{\{\ tc(T_{\neg CL \wedge \neg C \sim L})\ |\ L\ literal\ \}}$ of the above union is already implied by the left part $\mathbf{\{\ tc(T_{CL})\ |\ L\ literal\ \}}$.

Example 6.1 (temporal completion of a link formalization)

Let **T** be the temporal theory (a subset of $\mathbf{T_{DR}}$) that formalizes the information link diagnosis; see Example 4.4. The temporal completion of **T** contains the set of formulae (where $\mathbf{h \in HYPS}$):

$$\mathbf{[\ (\ C(DR\_uptodate(diagnosis))\ \wedge\ Y(output\_HV\_confirmed(h))\ )\ \vee\ (\ \neg(\ C(DR\_uptodate(diagnosis))\ \wedge\ Y(\neg\ output\_HV\_confirmed(h))\ )\ \wedge\ Y(output\_DR\_confirmed(h))\ )\ ]}$$

$$\mathbf{\leftrightarrow\quad C(output\_DR\_confirmed(h))}$$

$$\mathbf{[\ (\ C(DR\_uptodate(diagnosis))\ \wedge\ Y(\neg\ output\_HV\_confirmed(h))\ )\ \vee\ (\ \neg(C(DR\_uptodate(diagnosis))\ \wedge\ Y(output\_HV\_confirmed(h)))\ \wedge\ Y(\neg\ output\_DR\_confirmed(h))\ )\ ]}$$

$$\mathbf{\leftrightarrow\quad C(\neg\ output\_DR\_confirmed(h))}$$

Similarly formulae can be expressed for the case of rejected hypotheses. Note that the result of temporal completion is a temporal theory that is not anymore in executable format. However, the temporal completion allows to formalize proofs in a classical proof system; therefore Corollary 5.5 can be applied. For example, the collection of conservativity properties of Example 5.2 is both compositionally and globally provable with respect to $\vdash$ from the compositional theory $\mathbf{(tc(T_C))_{C \in COMP}}$, where each $\mathbf{T_C}$ is the temporal translation of the specification of component **C**.

The notion of temporal completion defined above expresses default persistence for all information in the system. This implies that in all cases where no default persistence is intended, explicit temporal rules are required that prohibit the persistence. For example, to describe retraction of information that deductively depends on other information that was revised (such as occurs, for example, in the truth maintenance process of primitive reasoning components in DESIRE), it is needed in addition to explicitly express a temporal rule, e.g., (for the Example 4.5) of the form:

$$\mathbf{(\ \phi \wedge \neg(\ Y(\neg\ input\_HD\_confirmed(h))\ \wedge\ Y(\neg\ input\_HD\_rejected(h))\ )\ ) \rightarrow \neg\ C(output\_HD\_focus(h))\ \wedge\ \neg\ C(\neg\ output\_HD\_focus(h))}$$

where ϕ is a formula expressing control information that allows the rule to be used (for example, the component should be active).

Application of Proposition 4.5 implies:

Corollary 6.2

For the language composition $\mathbf{(L_C)_{C \in COMP}}$, let a composition of properties $\mathbf{(P_C)_{C \in COMP}}$ and a compositional temporal theory $\mathbf{(T_C)_{C \in COMP}}$ be given. Let $\vdash$ be a classical provability relation for temporal epistemic logic.

If $\mathbf{(P_C)_{C \in COMP}}$ is compositionally provable with respect to $\vdash$ from the compositional temporal theory $\mathbf{(tc(T_C))_{C \in COMP}}$ then $\mathbf{(P_C)_{C \in COMP}}$ is globally provable with respect to $\vdash$ from the compositional theory $\mathbf{(tc(T_C))_{C \in COMP}}$.

Another approach is to define a more sensitive form of temporal completion already taking this into account, in which case these separate rules for retraction are not needed.

7 Conclusions

The usefulness of temporal epistemic logic was investigated to formalize verification proofs. As a test the properties that were found for verification of a diagnostic reasoning system were translated. It turns out that this logic is adequate if the executable temporal theories formalizing a specification are temporally completed. In this case classical provability can be used, which is much more transparent than the more complicated non-classical provability relations. Furthermore, our study shows that at least for the diagnostic example temporal epistemic logic provides enough expressivity for dynamics and reasoning about time, and formalizes incomplete information states in an adequate manner. To obtain the right structure in accordance with the compositional system design, in refinement to the logic a number of compositional structures were introduced: compositions of sub-languages, compositional theories, and compositional provability.

References

Abadi, M. and L. Lamport (1993). Composing Specifications, *ACM Transactions on Programming Languages and Systems*, Vol. 15, No. 1, 1993, pp. 73-132.

Barringer, H., M. Fisher, D. Gabbay, and A. Hunter (1991). Meta-Reasoning in Executable Temporal Logic, in: J. Allen, R. Fikes, E. Sandewall, *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, KR'91.

Barringer, H., M. Fisher, D. Gabbay, R. Owens, and M. Reynolds (1996). *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press Ltd. and John Wiley & Sons.

Benthem, J.F.A.K. van (1983). *The Logic of Time : a Model-theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*, Reidel, Dordrecht.

Brazier, F.M.T., J. Treur, N.J.E. Wijngaards, and M. Willems (1995). Formal Specification of Hierarchically (De)Composed Tasks. In: B.R. Gaines, M.A. Musen (Eds.), *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'95*, Calgary: SRDG Publications, Department of Computer Science, University of Calgary, pp. 25/1-15/20.

Brazier, F.M.T., J. Treur, N.J.E. Wijngaards, and M. Willems (1996). Temporal semantics of complex reasoning tasks. In: B.R. Gaines, M.A. Musen (Eds.), *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'96*, Calgary: SRDG Publications, Department of Computer Science, University of Calgary, 1996, pp. 15/1-15/17. Extended version to appear in: *Data and Knowledge Engineering*, 1998.

Cornelissen, F., C.M. Jonker, and J. Treur (1997). Compositional Verification of Knowledge-based Systems: a Case Study for Diagnostic Reasoning. *Proc. of the European Workshop on Knowledge Acquisition, Modelling and Management, EKAW'97*. Lecture Notes in AI, Springer Verlag.

Dams, D., R. Gerth, and P. Kelb (1996). *Practical Symbolic Model Checking of the full μ-calculus using Compositional Abstractions*. Report, Eindhoven University of Technology, Department of Mathematics and Computer Science.

Engelfriet, J. (1996). Minimal Temporal Epistemic Logic, *Notre Dame Journal of Formal Logic*, vol. 37, pp. 233-259 (special issue on Combining Logics).

Engelfriet, J., and J. Treur (1996a). Specification of Nonmonotonic Reasoning. *Proc. International Conference on Formal and Applied Practical Reasoning, FAPR'96*, Springer-Verlag, Lecture Notes in Artificial Intelligence, vol. 1085, pp. 111-125.

Engelfriet, J., and J. Treur (1996b). Executable Temporal Logic for Nonmonotonic Reasoning; *Journal of Symbolic Computation*, vol. 22, no. 5&6, pp. 615-625.

Engelfriet, J. and J. Treur (1997). An Interpretation of Default Logic in Temporal Epistemic Logic. *Journal of Logic, Language and Information*, to appear.

Finger, M. and D. Gabbay (1992). Adding a Temporal Dimension to a Logic System, *Journal of Logic, Language and Information* **1**, pp. 203-233.

Hooman, J. (1994). Compositional Verification of a Distributed Real-Time Arbitration Protocol. *Real-Time Systems*, vol. 6, pp. 173-206.

Jonker, C.M. and J. Treur (1998). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. In: H. Langmaack, A. Pnueli, W.P. De Roever (eds.). *Proceedings of the International Symposium on Compositionality, COMPOS'97*, Springer Verlag, to appear.

Treur, J., and M. Willems (1994). A logical foundation for verification. In: *Proceedings of the Eleventh European Conference on Artificial Intelligence, ECAI'94*, A.G. Cohn (Ed.), John Wiley & Sons, Ltd., pp. 745-749.