On compositionality and its limitations

Alexander Rabinovich
Tel Aviv University

The aim of this paper is to examine the applicability of a compositional method developed for a generalized product construction by Feferman and Vaught to the field of program verification.

We suggest an instance of the generalized product construction and prove an appropriate composition theorem for modal logic. We illustrate the usefulness of this generalized product by showing that many "parallel composition" operations are special cases of this generalized product.

We obtain positive results (the compositional method works) for basic propositional modal logic, and negative results (the compositional method fails) for more expressive logics which can express $\mathsf{EG}p$ -"there is a path such that all the nodes of the path have the property p".

Applications of the composition theorem to the model-checking problem and to the parametric model checking problem are provided.

Categories and Subject Descriptors: D.2.4 [**Software Engineering**]: Software/Program Verification[Formal methods, Model checking]; F.3.1 [**Logics and Meaning of Programs**]: Specifying and Verifying and Reasoning about Programs; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Modal Logic*

General Terms: Compositional Verification

Additional Key Words and Phrases: Feferman-Vaught theorem, composition methods, modal logic

1. INTRODUCTION AND SUMMARY OF RESULTS

The compositional approach reduces the verification of a property φ of a system $C(S_1,\ldots,S_n)$ assembled from the components $S_1,\ldots,S_n$ to the verification of other properties $\varphi_1,\ldots,\varphi_n$ of the components. There are two parameters here:

(1) The specification language L_{spec} in which properties are formulated.

(2) The collection of operations OP by which a complex system can be assembled from its components.

The ideal dream of compositionality (Composition Theorem) is to find an algorithm which for every formula $\varphi \in L_{spec}$ and every n-ary operator $C \in OP$ will construct formulas $\varphi_1,\ldots,\varphi_n$ such that $C(S_1,\ldots,S_n)$ satisfies φ iff S_1 satisfies φ_1, S_2 satisfies φ_2, ... and S_n satisfies φ_n.

Author's address: Alexander Rabinovich, Department of Computer Science, Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Israel 69978, e.mail: rabinoa@post.tau.ac.il.
The results presented in this paper were primarily obtained during 2000-2001 while the author was visiting the University of Edinburgh and was partially supported by the EPSRC fellowship GR/R03051.

In a seminal paper Feferman and Vaught [1959] introduced a generalized product of structures. The generalized product encompasses a great variety of algebraic constructions. The Feferman-Vaught composition theorem reduces the first-order theory of the generalized product to the first-order theory of the component structures (factors) and the monadic second-order theory of the index structure. In the Feferman-Vaught composition theorem L_{spec} is first-order logic and OP consists of all generalized products.

First-order logic is not very appropriate for the specification of concurrent and reactive systems because it distinguishes between bisimulation equivalent systems. (It is often argued that bisimulation invariant systems are indistinguishable for all reasonable notions of observation.) Multi-modal logic has the same expressive power as the bisimulation invariant fragment of first-order logic [van Benthem 1976].

We aim to explore the applicability of compositional methods in the area of verification. We will show that the composition theorem is realizable when the specification language L_{spec} is multi-modal logic and the set of operations OP consists of a wide variety of product ("parallel composition") operators. On the other hand, we will show if L_{spec} can express "there is a path such that all the nodes of the path have a property p", then (even non-algorithmic version of) the composition theorem fails for very simple parallel operators.

In recent years products of modal logics corresponding to products of Kripke frames were actively studied. Axiomatisation, decision and complexity problems for products of multi-modal logics were investigated (see a comprehensive survey by Gabbay and Shehtman [Gabbay and Shehtman 1998]).

Recall that a Kripke frame for basic modal logic is a structure $\mathcal{F} = \langle F,\ R\rangle$ where F is a set of states and R is a binary relation (the accessibility relation) on F. A Kripke structure over a frame $\mathcal{F} = \langle F,\ R\rangle$ is a structure $K = \langle F,\ R,\ P_1, \dots, P_i, \dots\rangle$ where P_i is a subset of F, which provides the interpretation of a propositional variable p_i in the structure. Reactive and concurrent systems such as computer hardware or software systems which exhibit nondeterministic behavior are typically represented by Kripke structures.

Suppose we need to combine two frames $\mathcal{F}_1 = \langle F_1,\ R_1\rangle$ and $\mathcal{F}_2 = \langle F_2,\ R_2\rangle$ into a complex frame $\mathcal{F}$. A natural way of combining is as follows. Let the states F of $\mathcal{F}$ be $F_1 \times F_2$ and define two accessibility relations on F: the first is for the accessibility relation of $\mathcal{F}_1$ where F_2 states are fixed and the second is for the accessibility relation of $\mathcal{F}_2$ where F_1 states are fixed, see Subsection 2.1 for a more detailed definition.

From the Computer Science point of view, it is more interesting to study products of Kripke structures instead of products of Kripke frames. Suppose we need to combine two Kripke structures $K_1 = \langle F_1,\ R_1, P^1_1, \dots, P^1_j, \dots\rangle$, $K_2 = \langle F_2,\ R_2, P^2_1, \dots, P^2_j, \dots\rangle$, into a complex Kripke structure. A natural way is to take the product of their frames $\langle F_1,\ R_1\rangle$ and $\langle F_2,\ R_2\rangle$ and then to specify how the interpretation of monadic predicates over $F_1 \times F_2$ is defined by the interpretations of the monadic predicates in K_1 and K_2 (see Subsection 2.1 for a more detailed definition).

In Subsection 2.2 we illustrate the composition theorem in a very simple context. As the set of operations we take the products of Kripke structures (these products correspond to parallel composition operations without communication). We will show (Theorem 2) that for the products of Kripke structures the composition theorem holds. Namely, Theorem 2 reduces the verification of a multi-modal formula φ over the products to the verification of modal properties (computable from φ) over the components of the product. A variant of this theorem was independently obtained by Gabbay and Shechtman [Gabbay and Shehtman 1999].

In Subsection 2.3 some simple applications of the composition theorem are provided. We will address two computational problems associated with the products. The first one is the Model Checking Problem (MCP) over a product σ: given a sequence $K_1 \ldots K_n$ of finite state Kripke structures, a state s in their σ product and a modal formula φ, determine whether the state s satisfies φ. The second problem is the satisfiability problem: given a formula φ and an n-ary product operation σ, determine whether there are Kripke structures $K_1, \ldots, K_n$ such that φ is satisfiable in their σ product.

As a consequence of the composition theorem we obtain an algorithm for the model checking problem over products of time complexity $O(g(|\varphi|) \times (|K_1| + |K_2| + \cdots + |K_n|))$ where g is a recursive function (we found only a non-elementary upper bound on g, see notes after Theorem 2). Hence, for MCP over products there is no need to construct the product Kripke structure which has size $|K_1| \times \cdots \times |K_n|$. Therefore, the state explosion problem for multi-modal logic can be avoided.

Another consequence of Theorem 2 is that the satisfiability problem over the product of Kripke structures is decidable.

In Subsection 2.4 we show that the composition theorem fails for simple product operations when multi-modal logic is replaced by a more expressive specification formalism. The expressive power of multi-modal logic is weak. It can only express local properties. In particular, there is no multi-modal formula which holds at a state s in a structure K iff there is a path from s such that all the nodes of the path have the property p. It is shown in Section 2.4 that the composition theorem fails for any logic which can express this property, however the composition theorem still holds for the extension of multi-modal logic by the reachability modality. The result provides a very sharp bound on the limitation of compositional methods.

In Section 3 we provide a definition of the generalized product of Kripke structures. The generalized product construct encompasses a wide variety of parallel composition operators. We state the composition theorem for multi-modal logics and generalized products. We also show that the composition theorem fails even over simple instances of the generalized product for any extension of multi-modal logics which can express reachability. In some sense the paper begins again in Section 3. The results obtained in the previous section are reproved in much more general framework. The product operations considered in Section 2 are almost trivial instances of the generalized product. We believe that for a didactic purpose it is helpful first to illustrate the compositional theorem and its consequence in the simplest framework (as was done is Section 2) and only afterwards to show the framework in its full generality (with many complex and non-trivial definitions).

In Section 4, we provide applications of the composition theorem of Section 3 for parametric model checking.

Section 5 concludes the paper.

2. COMPOSITION OF N-ARY PRODUCT

2.1 Preliminaries

The n-modal logic is propositional multi-modal logic with unary operators $\Diamond_1, \Diamond_2, \ldots, \Diamond_n$. Formulas of this language (n-modal formulas) are constructed from propositional variables by the Boolean operations and modalities. Frames for n-modal logic are structures of the form $\mathcal{F} = \langle F, R_1, \ldots, R_n \rangle$ where R_i (for $i = 1, \ldots, n$) is a binary relation on F. A Kripke model over an n-frame $\mathcal{F} = \langle F, R_1, \ldots, R_n \rangle$ is a structure $K = \langle F, R_1, \ldots R_n, P_1, \ldots P_i \ldots \rangle$ where P_i are subsets of F. F is the universe of the Kripke structure; the elements of F are called states or worlds; R_i is a binary (accessibility) relation, and P_i is the interpretation of a propositional variable p_i in the structure. The inductive definition of "formula φ is true at state s in a structure K" is the standard one, e.g., the clause for $\Diamond_i$ is as follows:

$$K, s \models \Diamond_i \varphi \text{ iff there is } s' \text{ such that } sR_i s' \text{ and } K, s' \models \varphi$$

Sometimes it is more convenient to index accessibility relations by the elements of a set A rather than by natural numbers. Hence, an A frame is a structure $\langle F, \{R_a : a \in A\} \rangle$ where R_a are binary relations on F; the elements of A are called actions. Similarly, monadic predicates on a frame are sometimes indexed by the elements of a set V. An (A, V) Kripke structure is a structure $\langle F, \{R_a : a \in A\}, \{P_v : v \in V\} \rangle$ where R_a are binary relations on F and P_v are subsets of F.

An (A, V) Kripke structure is finite if A, V and its set of states are all finite. The size of an (A, V) Kripke structure K is denoted by $|K|$ and is defined as usual. (For example, we can describe the transition relation of such an (A, V) Kripke structure as an array with one entry for every node; the entry for a node u contains a list of all pairs $\langle a, w \rangle$ such that $R_a(u, w)$; the size of K can be defined as the number of bits needed to describe the transition relation plus the number of bits needed to describe the subsets P_v for $v \in V$.) Whenever A and V are clear from the context or are irrelevant we use a "Kripke structure" for an "(A, V) Kripke structure".

The formulas of (A, V) modal logic (notation $\mathrm{ML}(A, V)$) are constructed from the set p_v ($v \in V$) of propositional variables by the Boolean operators and the unary modalities $\Diamond_a$ ($a \in A$). The semantics is defined as for n-modal logic.

PRODUCT OF FRAMES AND OF KRIPKE STRUCTURES. Given 1-frames $\mathcal{F}_1 = \langle S_1, R_1 \rangle, \ldots, \mathcal{F}_n = \langle S_n, R_n \rangle$, their product $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ is defined to be the n-frame $\langle S_1 \times \cdots \times S_n, \bar{R}_1, \ldots, \bar{R}_n \rangle$ where $\bar{R}_i$ is the following relation on $S_1 \times \cdots \times S_n$:

$$\langle s_1, \ldots, s_n \rangle \bar{R}_i \langle s'_1, \ldots, s'_n \rangle \text{ iff } s_i R_i s'_i \text{ and } s_j = s'_j \text{ for } j \neq i.$$

Let $K_1 = \langle S_1, R_1, P^1_1, \ldots, P^1_j, \ldots \rangle, \ldots, K_i = \langle S_i, R_i, P^i_1, \ldots, P^i_j, \ldots \rangle, \ldots, K_n = \langle S_n, R_n, P^n_1, \ldots, P^n_j, \ldots \rangle$ be Kripke structures over 1-frames $\mathcal{F} = \langle S_1, R_1 \rangle, \ldots, \mathcal{F}_i = \langle S_i, R_i \rangle, \ldots, \mathcal{F}_n = \langle S_n, R_n \rangle$. Their product is a Kripke structure over the frame $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$. The interpretation P_m of the propositional variable p_m in the product is defined by the interpretation of the propositional variables in the factor structures. A natural way to define an interpretation for p_k can be provided by a Boolean formula α_k over Boolean variables $\{p^j_i : j = 1, \ldots, n\}$. A state $\langle s_1, \ldots s_n \rangle$

will be in P_k iff α_k holds whenever p_i^j is defined as

$$p_i^j = \begin{cases} \text{True} & \text{if } s_j \in P_i^j \\ \text{False} & \text{otherwise} \end{cases}$$

Therefore, we define the product for Kripke structures over 1-frames as follows.

DEFINITION 1 n-ARY PRODUCT - SYNTAX. *An n-ary product operator is a function σ which assigns to variables p_k a Boolean formula $\alpha_{\sigma(p_k)}$ over variables $\{p_i^j : j = 1, \ldots, n\}$.*

The semantics of the product is defined as follows. Let σ be a product operator. Let $K_1 = \langle S_1,\ R_1,, P_1^1, \ldots P_j^1 \ldots \rangle$, $\ldots$, $K_n = \langle S_n,\ R_n,, P_1^n, \ldots P_j^n \ldots \rangle$ be Kripke structures over 1-frames. Their σ product $\Pi^\sigma_{i=1,\ldots,n} K_i$ is defined to be the structure $\langle S_1 \times \cdots \times S_n,\ \bar{R}_1, \ldots, \bar{R}_n, \ldots P_i \ldots \rangle$ where $\bar{R}_i$ $(i = 1, \ldots, n)$ is defined as in the product of 1-frames and P_k is the following unary relation on $S_1 \times \cdots \times S_n$: $\langle s_1, \ldots, s_n \rangle \in P_k$ if $\alpha_{\sigma(p_k)}$ holds under the interpretation of its Boolean variables as explained above.

The products of k-frames (and of Kripke structures over k-frames) are defined similarly to the products of 1-frames. For k-frames $\mathcal{F}_1, \ldots, \mathcal{F}_n$ their product $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ has $n \times k$ accessibility relations (one accessibility relation for each accessibility relation of every factor). In Subsections 2.2-2.4 the results are stated for Kripke structures over 1-frames. However, all the theorems and their proofs are easily extended to the products of Kripke structures over k-frames.

2.2 Composition Theorem for n-ary product

THEOREM 2 (COMPOSITION THEOREM FOR THE PRODUCT). *Let σ be an n-ary product operator. For every formula φ of n-modal logic there is a finite set I and 1-modal formulas φ_i^j $(i \in I,\ j = 1, \ldots, n)$ such that for every sequence of Kripke structures $K_1 = \langle S_1,\ R_i, P_1^1, \ldots, P_l^1, \ldots \rangle$, $\ldots$, $K_n = \langle S_n,\ R_n, P_1^n, \ldots, P_j^n, \ldots \rangle$ over 1-frames and every $\langle s_1, \ldots, s_n \rangle \in S_1 \times \cdots \times S_n$*

$$\Pi^\sigma_{i=1,\ldots,n} K_i, \langle s_1, \ldots, s_n \rangle \models \varphi$$

if and only if for some $i \in I$ and all $j = 1, \ldots, n$

$$K_j, s_j \models \varphi_i^j.$$

Moreover, the formulas φ_i^j are computable from φ and σ.

Remarks.

(1) The composition theorem reduces verification of φ in a product to a **finite** set of cases. For each case we only have to verify formulas on the components of the product.

(2) The best upper bound on the number of cases which we were able to extract from the proof below is $exp_{|\varphi|}(1)$, where $|\varphi|$ is the length of the formula φ and $exp_m(k)$ is m-time iterated exponential function (e.g., $exp_2(k) = 2^{2^k}$).

PROOF. We prove the theorem by induction on φ. For atomic φ it is clear from the definition and the observation that any Boolean formula over propositional variables $\{p_h^j : j = 1, \ldots, n\}$ is equivalent to a finite disjunction of the form $\bigvee_{i \in I}(\alpha_i^1 \wedge \alpha_i^2 \wedge \ldots \wedge \alpha_i^n)$ where α_i^j is a formula that contains only variables from $\{p_h^j : h \text{ - arbitrary}\}$.

For disjunction inductively assume that the theorem holds for ψ and θ. Let ψ_i^j $(i \in I_1)$ and θ_i^j $(i \in I_2)$ be the sequences of formulas correlated with ψ and θ. Without loss of generality we can assume that I_1 and I_2 are disjoint. Put $I = I_1 \cup I_2$ and define

$$\varphi_i^j = \begin{cases} \theta_i^j & \text{if } i \in I_1 \\ \psi_i^j & \text{if } i \in I_2 \end{cases}$$

It is easy to check that the theorem holds when we correlate φ_i^j $(i \in I_1 \cup I_2)$ with $\theta \vee \psi$.

For negation inductively assume that the theorem holds for ψ and let ψ_i^j $(i \in I)$ be a sequence of formulas correlated with ψ. Let $\mathcal{P}(I)$ be the set of all subsets of I. Let H be the set of functions from $\{1, 2, \ldots, n\}$ into $\mathcal{P}(I)$ defined as follows:

$$H \triangleq \{h \in \{1, 2, \ldots, n\} \to \mathcal{P}(I) \ : \ \forall i \in I \exists j \in \{1, 2, \ldots, n\}(i \notin h(j))\}.$$

For $h \in H$ define

$$\varphi_h^j \triangleq \bigwedge_{i \in h(j)} \psi_i^j \wedge \bigwedge_{i \notin h(j)} \neg \psi_i^j$$

It is easy to check that the theorem holds when we correlate φ_h^j $(h \in H)$ with $\neg\psi$.

Finally, for modality $\Diamond_m$, let ψ_i^j $(i \in I)$ be a sequence of formulas correlated with ψ. For $i \in I$ define

$$\varphi_i^j \triangleq \begin{cases} \psi_i^j & \text{if } j \neq m \\ \Diamond_m \psi_i^m & \text{if } j = m \end{cases}$$

It is easy to check that the theorem holds when we correlate φ_i^j $(i \in I)$ with $\Diamond_m \psi$. $\square$

2.3 Model Checking and satisfiability over products

Recall an n-ary product operator (see Definition 1) is defined by that a function σ. Throughout this section we will assume that σ is recursive (this is always the case for the structures with a finite number of accessibility relation names and monadic predicate names). Given an n-ary product operator σ, the *Model Checking problem over* σ is the following decision problem:

Input:. A sequence $K_1, \ldots, K_n$ of finite state Kripke structures, a sequence $s_1, \ldots, s_n$ of states (s_i is a state of K_i) and an n-modal formula φ.

Question:. Determine whether the state $\langle s_1, \ldots, s_n \rangle$ of $\Pi^\sigma K_i$ satisfies φ.

A naive algorithm for the model checking problem will first construct the product K of K_i and then evaluate the formula φ in K.

Note that the first step of this algorithm has space and time complexity $O(|K_1| \times \cdots \times |K_n|)$. Hence, even for a fixed formula φ this algorithm is exponential.

Recall the following theorem which was first proved in [Clarke and Emerson 1981]:

THEOREM 3. *There is an algorithm, that given a finite Kripke structure K, a state s of K and a modal formula φ decides whether $K, s \models \varphi$ in time $O(|K| \times |\varphi|)$.*

Theorem 3 together with the composition theorem implies the following result.

THEOREM 4. *The model checking problem over σ product can be decided in time $O(g(|\varphi|) \times (|K_1| + |K_2| + \cdots + |K_n|))$ where g is a recursive function.*

Now let us consider the satisfiability problem over products. First recall Theorem 5 from [Ladner 1977] and Theorem 6 from [Hirsch et al. 2002].

THEOREM 5. *The satisfiability problem for modal logic is PSPACE-complete.*

THEOREM 6 (SATISFIABILITY OVER THE PRODUCTS OF FRAMES IS UNDECIDABLE). *It is undecidable whether for a modal formula φ there are frames $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$ such that φ is satisfiable in a Kripke structure over the frame $\mathcal{F}_1 \times \mathcal{F}_2 \times \mathcal{F}_3$.*

Theorem 6 contrasts with the next theorem which follows from Theorem 5 and from the composition theorem.

THEOREM 7 (SATISFIABILITY OVER PRODUCTS OF KRIPKE STRUCTURES IS DECIDABLE). *Let σ be an n-ary product operator. There is an algorithm that decides whether an n-modal formula φ is satisfiable in the σ product of Kripke structures.*

Let $C_1, \dots, C_n$ be classes of Kripke structures. The σ product $\Pi^\sigma C_i$ of $C_1, \dots, C_n$ is the class $\{\Pi^\sigma K_i \ : \ K_i \in C_i \text{ for } i = 1, \dots, n\}$ of Kripke structures. Consider the following refinements of the satisfiability problem.

Satisfiability problem over $\Pi^\sigma C_i$: Given an n-modal formula φ. Decide whether φ is satisfiable in the class $\Pi^\sigma C_i$.

From the composition theorem it follows

COROLLARY 8. *The satisfiability problem over the class $\Pi^\sigma C_i$ of structures is recursively reducible to the satisfiability problems over the classes $C_1, \dots, C_n$.*

PROOF. Given a formula φ. Construct formulas φ_i^j which correspond to φ by Theorem 2. The formula φ is satisfiable in $\Pi^\sigma C_i$ if and only if there is i such that for every $j = 1, \dots, n$ the formula φ_i^j is satisfiable over the class C_j. $\square$

2.4 Composition Theorem Fails for Expressive Logics

The expressive power of multi-modal logic is weak. It can only express local properties. Recall that a partial path in a structure K is a (finite or infinite) sequence $s_0, \dots, s_i, \dots$ of nodes such that $\forall i \exists j (s_i R_j s_{i+1})$. A state s' is at distance $\leq d$ from s if there is a partial path of length $\leq d$ which starts at s and ends at s'. For every modal formula φ there is a number d(= the modal depth of φ) such that the truth value of φ at a state s of K is determined by the substructure of K over the states at the distance at most d from s.

Recall that a path is a partial path which is either infinite or it is finite and no node is accessible from its last state. From the above observation it follows that none of the following properties (of a state s of a Kripke structure) is expressible in multi-modal logic:

EFp:. there is a finite partial path which starts at s such that its last node has the property p. In other words, **EF**p holds at s iff a node with property p is reachable from s.

EGp:. there is a path which starts at s such that all the nodes of the path have the property p.

p **UNTIL** q:. there is a partial path which starts at s such that its last node has the property q and all the other nodes have the property p.

The next theorem shows that the composition theorem holds for the extension of n-modal logic with the modality **EF**. However, the main result of this subsection (Theorem 11) states that the composition theorem fails for any logic which can express the property **EG**p. (A similar result holds for the property p **UNTIL** q.)

THEOREM 9 (COMPOSITION THEOREM FOR **EF**). *Let σ be an n-ary product operator. For every formula φ of the extension of n-modal logic by the modality* **EF** *there is a finite set I and formulas φ_i^j $(i \in I,\ j = 1, \dots, n)$ in the modal logic with the modalities $\Diamond_1$ and* **EF** *such that for every sequence of Kripke structures $K_1 = \langle S_1,\ R_i, P_1^1, \dots, P_j^1, \dots\rangle$, $\dots$, $K_n = \langle S_n,\ R_n, P_1^n, \dots, P_j^n, \dots\rangle$ over 1-frames and every $\langle s_1, \dots, s_n\rangle \in S_1 \times \dots \times S_n$*

$$\Pi^{\sigma}_{i=1,\dots,n} K_i, \langle s_1, \dots, s_n\rangle \models \varphi$$

if and only if for some $i \in I$ and all $j = 1, \dots, n$

$$K_j, s_j \models \varphi_i^j.$$

Moreover, the formulas φ_i^j are computable from φ and σ.

PROOF. We prove the theorem by induction on φ. The case of atomic formulas and the inductive steps for disjunction, negation and $\Diamond_i$ is exactly like in the proof of Theorem 2.

The case of **EF** is treated as follows. Let ψ_i^j $(i \in I)$ be a sequence of formulas correlated with ψ. For $i \in I$ define

$$\varphi_i^j \triangleq \mathbf{EF}\psi_i^j$$

It is easy to check that the inductive assertion holds when we correlate φ_i^j $(i \in I)$ with $\mathbf{EF}\psi$. $\square$

Now we are going to show that the composition theorem fails for any logic which can express **EG**p.

The idea of the proof is as follows. We define a formula ψ, a binary product operator $\times^{\sigma}$ and an infinite family $\{C_j\ :\ j \in \mathbf{Nat}\}$ of Kripke structures with a common state s_0 such that

$$\text{The state } \langle s_0, s_0\rangle \text{ of } C_i \times^{\sigma} C_j \text{ satisfies } \psi \text{ if and only if } i = j. \qquad (1)$$

From (1), the failure of the composition theorem for any logic L that can express ψ is derived as follows. For the purpose of the contradiction assume that the composition theorem holds for L. Then there is a finite family of formulas ϕ_1^i, ϕ_2^i $(i \in I)$ such that

$$\begin{gathered} K_1 \times^{\sigma} K_2, \langle s_1, s_2\rangle \models \psi \text{ if and only if} \\ K_1, s_1 \models \phi_1^i \text{ and } K_2, s_2 \models \phi_2^i \text{ for some } i \in I \end{gathered} \qquad (2)$$

Define an equivalence relations on $\{C_j\ :\ j \in \mathbf{Nat}\}$ as follows: C_j and $C_{j'}$ are *equivalent* iff s_0 in C_j and s_0 in $C_{j'}$ satisfy the same formulas from ϕ_1^i $(i \in I)$, i.e. $C_j, s_0 \models \phi_1^i \iff C_{j'}, s_0 \models \phi_1^i$ for all $i \in I$.

Since ϕ_1^i $(i \in I)$ is a finite set of formulas, the above equivalence partitions the infinite set $\{C_j\ :\ j \in \mathbf{Nat}\}$ of Kripke structures into a finite set of equivalence classes. Hence, there is a non-singular equivalence class, i.e. an equivalence class that contains at least two elements $C_j, C_{j'}$ for $j \neq j'$ (actually there is a class that contains an infinite number of different elements).

From (1) we have that $C_j \times^\sigma C_j, \langle s_0, s_0\rangle \models \psi$ and $C_{j'} \times^\sigma C_j, \langle s_0, s_0\rangle \models \neg\psi$. However, from (2) and the fact that $\langle s_0, C_j\rangle$ and $\langle s_0, C_{j'}\rangle$ are indistinguishable by formulas from ϕ_1^i $(i \in I)$ it follows that $C_j \times^\sigma C_j, \langle s_0, s_0\rangle \models \psi$ iff $C_{j'} \times^\sigma C_j, \langle s_0, s_0\rangle \models \psi$. Contradiction.

Actually, the above arguments can be easily modified for weaker assumptions: It is sufficient to provide two infinite families of distinct structures $\{C_j \ : \ j \in \textsf{Nat}\}$ and $\{D_j \ : \ j \in \textsf{Nat}\}$ and replace (1) by the condition "$C_i \times^\sigma D_j, \langle s_0, s_0\rangle \models \psi$ if and only if $i = j$."

Now we are going to complete the above sketch by providing the appropriate details. We are going to define an infinite family of structures C_i and an appropriate binary product $\times^\sigma$.

Consider a Kripke structure $C_n = \langle S,\ R,\ P_0,\ P_1,\ P_2\rangle$ defined as follows:

States:. the universe S is $\{1, 2, 3, \ldots, 3n\}$;

Accessibility relation:. R is interpreted as the successor relation on S, i.e., $R = \{\langle i,\ i+1\rangle \ : \ i = 1, \ldots, 3n-1\}$;

Monadic Predicates:. $P_0 = \{i \in S \ : \ i \ mod\ 3 = 0\}$, $P_1 = \{i \in S \ : \ i \ mod\ 3 = 1\}$ and $P_2 = \{i \in S \ : \ i \ mod\ 3 = 2\}$.

Consider a binary product σ which defines P on the product as follows: $\langle s_1, s_2\rangle$ is in P iff

s_1 has the property p_0 and s_2 has the property $p_0 \vee p_1$ or
s_1 has the property p_1 and s_2 has the property $p_1 \vee p_2$ or
s_1 has the property p_2 and s_2 has the property $p_2 \vee p_0$.

The reader is invited to write down the corresponding formal definition for this product. Figure 1 shows the product of $C_2 \times^\sigma C_3$. The following lemma is immediate.

Lemma 10. *The state $\langle 1, 1\rangle$ of $C_n \times^\sigma C_m$ satisfies $\textsf{EG}p$ if and only if $n = m$.*

Proof. First observe that the node $\langle 3n, 3m\rangle$ is the last node in every (full) path in the structure $C_n \times^\sigma C_m$.

Note also that the set H of nodes reachable from $\langle 1, 1\rangle$ by a partial path with all nodes having property p is

$$H = \{\langle i, i\rangle \ : \ i \leq 3 \times min(n, m)\} \cup \{\langle i, i+1\rangle \ : \ i < 3 \times min(n, m)\}$$

Therefore, if $n \neq m$ there is no (full) path from $\langle 1, 1\rangle$ such that all the nodes on the path have a property p.

Theorem 11 (The composition theorem fails for expressive logics). *Let L be any logic which can express $\textsf{EG}p$. There is no finite set I and a sequence of L-formulas $\phi_1^i,\ \phi_2^i$ $(i \in I)$ such that*

$$K_1 \times^\sigma K_2, \langle s_1, s_2\rangle \models \textsf{EG}p \textit{ if and only if}$$

$$K_1, s_1 \models \phi_1^i \textit{ and } K_2, s_2 \models \phi_2^i \textit{ for some } i \in I$$

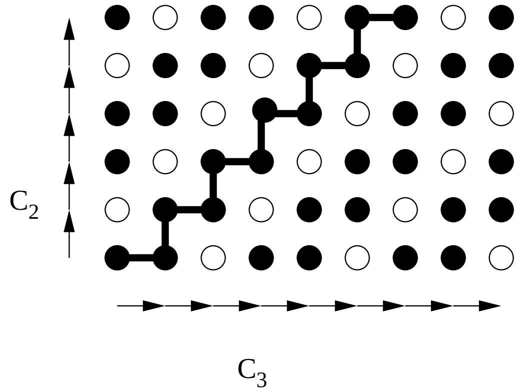

Fig. 1. The p-property nodes of $C_2 \times^\sigma C_3$ are drawn in black.The partial path whose all nodes have a property p is drawn.

PROOF. For contradiction assume that there are $\phi_1^i,\ \phi_2^i$ such that

$$\begin{array}{c} K_1 \times^\sigma K_2, \langle s_1, s_2\rangle \models \mathbf{EG}\mathrm{p} \\ \text{if and only if} \\ K_1, s_1 \models \phi_1^i \text{ and } K_2, s_2 \models \phi_2^i \text{ for some } i \in I \end{array} \tag{3}$$

For $i \in I$ let N_i be defined as

$$N_i = \{n \in \mathbf{Nat}\ :\ C_n, 1 \models \phi_1^i \text{ and } C_n, 1 \models \phi_2^i\}, \tag{4}$$

Since the state $\langle 1, 1\rangle$ of $C_n \times^\sigma C_n$ satisfies $\mathbf{EG}$p (see Lemma 10), it follows from (3) that for every $n \in \mathbf{Nat}$ there is $i \in I$ such that $n \in N_i$.

Recall that I is finite; therefore, there is $i_0 \in I$ such that N_{i_0} contains at least two elements. Let $n_1 \neq n_2$ be two elements of N_{i_0}. Observe that $C_{n_1}, 1 \models \phi_1^{i_0}$ and $C_{n_2}, 1 \models \phi_2^{i_0}$ by (4) and the choice of i_0. Therefore, by (3), we obtain that $C_{n_1} \times^\sigma C_{n_2}, \langle 1, 1\rangle \models \mathbf{EG}$p. Therefore, (by Lemma 10) $n_1 = n_2$. Contradiction. □

Sometimes in the literature the variable free fragment of multi-modal logic is considered. This fragment is often called Hennessy-Milner Logic [Hennessy and Milner 1985]. In this fragment the formulas are constructed from True and False by the Boolean operations and modalities. Such formulas are interpreted over Kripke frames (since there is no need for an interpretation of variables). We will show below that the composition theorem fails over the product of frames for any logic that contains the variable free fragment of multi-modal logic and modality **EG**.

Consider Kripke structures $B_n = \langle S_n,\ R_b,\ R_0,\ R_1,\ R_2\rangle$ and $D_n = \langle S_n,\ R_d,\ R_3,\ R_4,\ R_5\rangle$ defined as follows (see Figure 2):

States:. $S_n = \{1, 2, 3, \ldots, 6n\}$.

Accessibility relations:. (1) $R_b = R_d = \{\langle i,\ i+1\rangle\ :\ i = 1, \ldots, 3n-1\}$;
(2) $R_0 = R_3 = \{\langle i,\ 3n+i\rangle\ :\ \text{where } i \leq 3n \text{ and } i \ mod\ 3 = 0\}$;
(3) $R_1 = R_4 = \{\langle i,\ 3n+i\rangle\ :\ \text{where } i \leq 3n \text{ and } i \ mod\ 3 = 1\}$;
(4) $R_2 = R_5 = \{\langle i,\ 3n+i\rangle\ :\ \text{where } i \leq 3n \text{ and } i \ mod\ 3 = 2\}$;

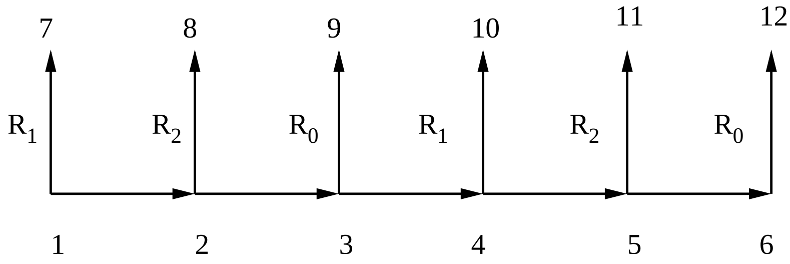

Fig. 2. B_2

Let ψ be defined as

$$\begin{aligned}\psi \triangleq\ & \big(\Diamond_0\mathrm{True} \wedge (\Diamond_3\mathrm{True} \vee \Diamond_4\mathrm{True})\big)\\ & \vee\big(\Diamond_1\mathrm{True} \wedge (\Diamond_4\mathrm{True} \vee \Diamond_5\mathrm{True})\big)\\ & \vee\big(\Diamond_2\mathrm{True} \wedge (\Diamond_5\mathrm{True} \vee \Diamond_3\mathrm{True})\big)\end{aligned}$$

This is a variable free formula. Let $E_{n.m}$ be the product of Kripke frames B_n and D_m (this is a frame with eight accessibility relations). It is easy to show that the state $\langle 1, 1\rangle$ of $E_{n,m}$ satisfies $\mathbf{EG}\psi$ if and only if $n = m$. Hence, by the same arguments as in the proof of Theorem 11 one can derive that the composition theorem fails over the product of frames for any logic that contains the variable free fragment of multi-modal logic and modality **EG**.

3. GENERALIZED PRODUCT OF KRIPKE STRUCTURES

In Section 2.2 we considered the composition theorem for multi-modal logic and very simple product operations which correspond to parallel composition without communication. In Section 2.4 we showed that the composition theorem fails even for these simple product operations when the multi-modal logic is replaced by a more expressive specification formalism. Here we will show that the composition theorem holds for multi-modal logic and a wide variety of product operations which cover most parallel composition operators considered in the literature.

In subsection 3.2 we give a definition of the generalized product of Kripke structures and in subsection 3.3 we prove the corresponding composition theorem for modal logics. In subsection 3.4 we show that the composition theorem fails over the generalized products and any logic which has the reachability modality **EF**. In Section 4 we derive some consequences of composition theorem for parametric model checking.

The generalized product construct encompasses a wide variety of ways to assemble a complex system from its components (factors). Henceforth, before providing the definition we consider some of its instances.

3.1 Examples of Products

In this subsection we present many "parallel composition" operators that can be found in the literature on concurrency or in the formalizations of distributed systems. All these "parallel compositions" are instances of the generalized product which will be presented in the next subsection.

In the following examples we assume that accessibility relations are indexed by a set A of actions; the frames with accessibility relations indexed by a set A will be called A-frames or frames over A.

Example 12 (Synchronous Product). Given A-frames $\mathcal{F}_1 = \langle S_1,\ \{R_a^1\ :\ a \in A\}\rangle, \ldots, \mathcal{F}_n = \langle S_n,\ \{R_a^n\ :\ a \in A\}\rangle$, their synchronous composition is defined as the A-frame $\langle S_1 \times \cdots \times S_n,\ \{R_a^s\ :\ a \in A\}\rangle$, where R_a^s (for $a \in A$) is the following relation on $S_1 \times \cdots \times S_n$:

$$\langle s_1, \ldots, s_n\rangle R_a^s \langle s_1', \ldots, s_n'\rangle \text{ iff } s_i R_a^i s_i' \text{ for all } i.$$

Remark 13. Some explanations about the notations and terminology used here might be helpful for the reader who is used to the notations of concurrency theory. What we call an A-frame is called a labeled transition system over the alphabet A. Instead of $sR_a s'$ the notations $s \rightarrow_a s'$ are used in the literature. Usually in concurrency, systems are described by process expressions. The labeled transition system is associated with the set of process expressions; the states of this transition system are the process expressions, and the transition relations $\rightarrow_a$ on expressions are defined by appropriate transition rules. For example, the rule for the synchronous product ($Synch$) above would be

$$\frac{E_1 \rightarrow_a E_1',\ E_2 \rightarrow_a E_2', \ldots, E_n \rightarrow_a E_n'}{Synch(E_1, \ldots, E_n) \rightarrow_a Synch(E_1', \ldots, E_n')}$$

Throughout this subsection we just recall some operations considered in the literature and restate (in a straigtforward way) transition rules for these operations in the logical notations which are used in the definition of the generalized product in Subsection 3.2.

Example 14 (Shuffle). The asynchronous composition (shuffle) of A-frames $\mathcal{F}_1 = \langle S_1,\ \{R_a^1\ :\ a \in A\}\rangle, \ldots, \mathcal{F}_n = \langle S_n,\ \{R_a^n\ :\ a \in A\}\rangle$ is defined as the A-frame $\langle S_1 \times \cdots \times S_n,\ \{R_a^{shuf}\ :\ a \in A\}\rangle$, where R_a^{shuf} (for $a \in A$) is the following relation on $S_1 \times \cdots \times S_n$:

$$\langle s_1, \ldots, s_n\rangle R_a^{shuf} \langle s_1', \ldots, s_n'\rangle \text{ iff there is } i \text{ such that } s_i R_a^i s_i' \text{ and } s_j = s_j' \text{ for } j \neq i.$$

Remark 15 (Shuffle vs. the product of frames). Let $\mathcal{F}_1 = \langle S_1,\ R^1\rangle, \ldots, \mathcal{F}_n = \langle S_n,\ R^n\rangle$ be one-frames. Their product (see Subsection 2.1) is n-frame $\langle S_1 \times \cdots \times S_n,\ \bar{R}^1, \ldots, \bar{R}^n\rangle$. However, their shuffle is one-frame $\langle S_1 \times \cdots \times S_n,\ R^{shuf}\rangle$ where the accessibility relation R^{shuf} is the union of $\bar{R}^i$.

In the following examples a set A of actions will be structured.

Example 16 (CCS parallel composition [Milner 1989]). Let Σ be a set (of input communication actions) and let $\bar{\Sigma} = \{\bar{c}\ :\ c \in \Sigma\}$ be a set (of output actions). Assume that sets Σ and $\bar{\Sigma}$ are disjoint and $\tau \notin \Sigma \cup \bar{\Sigma}$ (τ is called an internal or invisible action). For $d = \bar{c} \in \bar{\Sigma}$ we define $\bar{d}$ as c. A set of actions A is $\Sigma \cup \bar{\Sigma} \cup \{\tau\}$. Let $\mathcal{F}_1 = \langle S_1,\ \{R_a^1\ :\ a \in A\}\rangle$ and $\mathcal{F}_2 = \langle S_2,\ \{R_a^2\ :\ a \in A\}\rangle$ be A-frames. Their CCS composition is defined as the A-frame $\langle S_1 \times S_2,\ \{R_a\ :\ a \in A\}\rangle$,where the relation R_c for $c \in \Sigma \cup \bar{\Sigma}$ is defined as

$$\langle s_1, s_2\rangle R_c \langle s_1', s_2'\rangle \text{ iff either } s_1 R_c s_1' \text{ and } s_2 = s_2' \text{ or } \ s_2 R_c s_2' \text{ and } s_1 = s_1',$$

and R_τ is defined as

$$\langle s_1, s_2\rangle R_\tau \langle s_1', s_2'\rangle \text{ iff } \begin{cases} \text{for some } c \in \Sigma \cup \bar{\Sigma}, \quad s_1 R^1_c s_1' \text{ and } s_2 R^2_{\bar{c}} s_2' \text{ or} \\ s_1 R^1_\tau s_1' \text{ and } s_2 = s_2' \text{ or} \\ s_2 R^2_\tau s_2' \text{ and } s_1 = s_1'. \end{cases}$$

Example 17 (Broadcast Composition). In broadcast protocols [Emerson and Namjoshi 1996; 1998; Esparza et al. 1999] a set of actions A is composed of a set Σ_l of local actions, two sets $\Sigma_r \times \{?\}$ and $\Sigma_r \times \{!\}$ of input and output rendezvous actions and two sets $\Sigma_b \times \{??\}$ and $\Sigma_b \times \{!!\}$ of input and output broadcast actions.

Let $\mathcal{F}_i$ be A-frames. Their broadcast composition is a frame $\mathcal{F}$ over action set $\Sigma_l \cup \Sigma_b \cup \Sigma_r$. Frames $\mathcal{F}_i$ are called the components and $\mathcal{F}$ is called the systems defined by the broadcast composition.

The system performs a local action c if one of its components performs c and moves to a new state and the other components do not change their state. The system performs a rendezvous action $c \in \Sigma_r$ if two components perform a rendezvous, (i.e., one performs $c!$ transition and the second one performs $c?$ transition) and the other components do not change their state. The system performs a broadcast action $c \in \Sigma_b$ if one of its components performs $c!!$ transition and all the other components perform $c??$ transitions.

A product appropriate for broadcast protocols can be formalized as follows. Let $\mathcal{F}_i = \langle S_i,\ \{R^i_a \ :\ a \in A\}\rangle$ $(i \in I)$ be a family of frames over $A = \Sigma_l \cup \Sigma_b \times \{??\} \cup \Sigma_b \times \{!!\} \cup \Sigma_r \times \{?\} \cup \Sigma_r \times \{!\}$. Their (broadcast) product is defined as the following frame $\langle S,\ \{R_a \ :\ b \in B\}\rangle$ over $B = \Sigma_l \cup \Sigma_b \cup \Sigma_r$, where the set S of states is the Cartesian product of the sets S_i $(i \in I)$ and the relations R_c are defined as follows:

For $c \in \Sigma_l$

$$\langle \ldots, s_i, \ldots\rangle R_c \langle \ldots, s_i', \ldots\rangle \text{ iff there is } i \text{ such that } s_i R_c s_i' \text{ and } s_j = s_j' \text{ for } j \neq i.$$

For $c \in \Sigma_r$

$$\langle \ldots, s_i, \ldots s_m \ldots\rangle R_c \langle \ldots, s_i', \ldots s_m' \ldots\rangle \text{ iff there are } i \text{ and } m \text{ such that } i \neq m \text{ and } \\ s_i R_{c?} s_i' \text{ and } s_m R_{c!} s_m' \text{ and } s_j = s_j' \text{ if } j \neq i \wedge j \neq m.$$

For $c \in \Sigma_b$

$$\langle \ldots, s_i, \ldots\rangle R_c \langle \ldots, s_i', \ldots\rangle \text{ iff there is } i \text{ such that } s_i R_{c!!} s_i' \text{ and } s_j R_{c??} s_j' \text{ for } j \neq i.$$

Example 18 (Network Composition). In the previous examples we dealt with families $\mathcal{F}_i$ $(i \in I)$ of frames over an index set I. No structure on the index set I was assumed. The next example of a communication network assumes that we have an index structure $Ind = \langle I, Edge\rangle$ where I is a set and $Edge$ is a binary relation on I. Let Ind be such a structure and let $\mathcal{F}_i = \langle S_i,\ \{R^i_a \ :\ a \in A\}\rangle$ $(i \in I)$ be a family of frames over $A = \Sigma_l \cup \Sigma_r \times \{?\} \cup \Sigma_r \times \{!\}$. The network product of $\mathcal{F}_i$ over Ind is defined as the following frame $\langle S,\ \{R_a \ :\ b \in B\}\rangle$ over $B = \Sigma_l \cup \Sigma_r$, where the set S of states is the Cartesian product of the sets S_i $(i \in I)$ and for $c \in \Sigma_l \cup \Sigma_r$, the relations R_c are defined as follows:

For $c \in \Sigma_l$

$$\langle \ldots, s_i, \ldots\rangle R_c \langle \ldots, s_i', \ldots\rangle \text{ iff there is } i \text{ such that } s_i R_c s_i' \text{ and } s_j = s_j' \text{ for } j \neq i.$$

For $c \in \Sigma_r$

$$\langle \ldots, s_i, \ldots s_m \ldots \rangle R_c \langle \ldots, s'_i, \ldots s'_m \ldots \rangle \text{ iff there are } i \neq m \text{ such that } Edge(i,m) \text{ and } s_i R_{c?} s'_i \text{ and } s_m R_{c!} s'_m \text{ and } s_j = s'_j \text{ if } j \neq i \wedge j \neq m.$$

In other words, a system performs a local action c if one of its components performs c and moves to a new state and the other components do not change their state. A system performs a rendezvous action c if two components i and j connected by the edge from i to j perform a rendezvous, (i.e., i performs $c!$ transition and j performs $c?$ transition) and the other components do not change their state.

3.2 Generalized Product

In this section we introduce the generalized product of Kripke structures. It is a proper instance of Feferman-Vaught generalized product construct [Feferman and Vaught 1959]. However, the Feferman-Vaught product deals with arbitrary first-order languages and first-order structures. The product introduced here deals with Kripke structures and multi-modal logic. First-order logic is not very appropriate for the specification of concurrent and reactive systems because it distinguishes between bisimulation equivalent systems. Multi-modal logic has the same expressive power as the bisimulation invariant fragment of first-order logic [van Benthem 1976].

Let τ be a signature (i.e., a set of predicate and function symbols). We use $MSO(\tau)$ for the monadic second-order language over τ, i.e., $MSO(\tau)$ is the extension of the first-order language over τ by monadic predicate variables (set variables) and by the quantification over these variables. We use upper case letters for the monadic variables and lower case letters for the first-order variables.

Let K_i $(i \in I)$ be a family of Kripke structures with accessibility relations indexed by the elements of a set A and propositional variables indexed by the elements of a set V (the same sets A and V for all the structures). We use $ML(A,V)$ for the modal logic appropriate for these structures.

An $(A, V, \tau_{\text{index}})$ **determining sequence for an unary predicate** is a finite sequence of the form $\langle \alpha_1, \ldots, \alpha_n; \beta(X_1, \ldots, X_n) \rangle$ where α_i $(i = 1, \ldots, n)$ are formulas in $ML(A,V)$ and $\beta(X_1, \ldots, X_n)$ is a formula in the monadic second order logic for the signature τ_{index}, and β has no free first-order variables and it has one free monadic variable X_i for every formula α_i $(i = 1, \ldots, n)$.

An $(A, V, \tau_{\text{index}})$ **determining sequence for an accessibility relation** consists of

(1) a finite sequence $a_1, \ldots, a_m$ of distinct actions in A.
(2) a finite sequence $\alpha_1, \ldots, \alpha_n$ of formulas in $ML(A,V)$.
(3) a formula $\beta(Z_1, \ldots, Z_m, X_1, \ldots, X_n)$ in the monadic second-order logic for the signature τ_{index} where β has no free first-order variables and it has one free monadic variable Z_j for every action a_j $(j = 1, \ldots, m)$ that appears in (1) and one free monadic variable X_i for every formula α_i $(i = 1, \ldots, n)$; no other variable is free in β.

We use the notation $\langle a_1, \ldots, a_m; \alpha_1, \ldots \alpha_n; \beta \rangle$ for such sequences. If $n = 0$ we write $\langle a_1, \ldots, a_m; \ ; \beta \rangle$.

DEFINITION 19 (GENERALIZED PRODUCT OF KRIPKE STRUCTURES - SYNTAX). *A generalized product operation is a tuple* $\sigma = \langle A,\ V_c,\ B,\ V,\ \tau_{index},\ \rho \rangle$ *where*

—*A is a set of actions (the actions of the component Kripke structures).*

—*V_c is a set of indexes for the variables (of the component Kripke structures).*

—*B is a set of actions (the actions of the product Kripke structures).*

—*V is a set of indexes for the variables (of the product Kripke structures).*

—*τ_{index} is a signature for index structures and*

—*ρ is a function that assigns to every $b \in B$ an $(A,\ V_c,\ \tau_{index})$ determining sequence for an accessibility relation and to every $p \in V$ an $(A,\ V_c,\ \tau_{index})$ determining sequence for a unary predicate.*

SEMANTICS. The semantics of a product operation σ is defined as follows. Let I be a set and let Ind be a τ_{index} structure over the universe I. Let $K_i = \langle S_i, \{R^i_a\ : a \in A\},\ \{P^i_v\ :\ v \in V_c\}\rangle$ $(i \in I)$ be a family of $(A,\ V_c)$ Kripke structures. The σ product of K_i over Ind is a $(B,\ V)$ Kripke structure $K = \langle S, \{R_b\ : b \in B\},\ \{P_v\ :\ v \in V\}\rangle$ which is defined as follows.

States:. S is the Cartesian product of the family S_i $(i \in I)$ of sets; hence, the set of states is the set of all the functions g with the domain I such that for each $i \in I$, $g(i)$ is an element of S_i.

Interpretation of unary predicates:. P_v $(v \in V)$ is a unary relation on S defined as follows: let $\langle \alpha_1, \dots \alpha_n;\ \beta(X_1, \dots, X_n)\rangle$ be the determining sequence assigned by ρ to v. For $g \in S$ and $l = 1, \dots, n$ let I^g_l be the set $\{i \in I\ :\ K_i, g(i) \models \alpha_l\}$. Then $P_v \triangleq \{g \in S\ :\ Ind \models \beta(I^g_1, I^g_2, \dots, I^g_n)\}$. (We say that P_v is defined by the sequence $\langle \alpha_1, \dots \alpha_n;\ \beta(X_1, \dots, X_n)\rangle$ or that $\langle \alpha_1, \dots \alpha_n;\ \beta(X_1, \dots, X_n)\rangle$ defines P_v.)

Interpretation of accessibility relations:. R_b $(b \in B)$ is an accessibility relation on S defined as follows: let $\langle a_1, \dots, a_m;\ \alpha_1, \dots \alpha_n;\ \beta(Z_1, \dots, Z_m, X_1, \dots, X_n)\rangle$ be the determining sequence assigned by ρ to b. For $g \in S$ and $l = 1, \dots, n$ let $I^g_l \subseteq I$ be defined as above. Then gR_bg' iff there are disjoint sets $J_r \subseteq I$ $(r = 1, \dots, m)$ such that

(1) $Ind \models \beta(J_1, \dots, J_m, I^g_1, \dots, I^g_n)$ and
(2) $g(i)R_{a_l}g'(i)$ for every $l = 1, \dots, m$ and $i \in J_l$ and
(3) $g(i) = g'(i)$ for $i \in I \setminus \bigcup_{l=1}^{r} J_l$.
(We say R_b is defined by the sequence $\langle a_1, \dots, a_m;\ \alpha_1, \dots \alpha_n;\ \beta(Z_1, \dots, Z_m, X_1, \dots, X_n)\rangle$ or that this sequence defines R_b.)

Let us illustrate these definitions by examples.

EXAMPLE 20. *(1) Assume that the unary predicate P is defined by the determining sequence $\langle P_1; \forall t.\ t \in X_1\rangle$. Then $g \in S$ satisfies P iff for all i the i-th component of g is in a state that satisfies P_1.*

(2) Assume that Q is defined by the determining sequence $\langle P_1, P_2;\ \exists! t.(t \in X_1 \vee t \in X_2)\rangle$ where "$\exists!$" stands for "there is a unique". Then $g \in S$ satisfies Q iff exactly one of the $g(i)$ is in a state that satisfies P_1 or in a state that satisfies P_2.

(3) The accessibility relation R_a $(a \in A)$ for the synchronous product of A-frames is defined by the determining sequence $\langle a;\ ; \beta(Z_1)\rangle$; this sequence does not contain any α formula and $\beta(Z_1)$ is $\forall t.\ t \in Z_1$. Similarly, the accessibility relation R_a $(a \in A)$ of the shuffle of A-frames is defined by the sequence $\langle a;\ ;\ \exists! t.\ t \in Z_1\rangle$.

(*4*) *For the CCS parallel composition [Milner 1989], the accessibility relation is defined by the following sequences: for* $c \in \Sigma \cup \bar{\Sigma}$*, the determining sequence is* $\langle c\ ;\ ; \exists! t_1 .\ (t_1 \in Z_1)\rangle$*. In the case when* Σ *is a finite set* $\{a_1, \ldots, a_m\}$ *the determining sequence for* R_τ *is* $\langle a_1, \ldots, a_m, \bar{a}_1, \ldots, \bar{a}_m, \tau;\ ;\ \beta\rangle$*, where* $\beta(Z_1, \ldots, Z_{2m+1})$ *says that either for* $i \le m$ *the sets* Z_i *and* Z_{m+i} *have only one element and all the other sets* Z_j *are empty* $(j \notin \{i, i+m\})$ *or* Z_{2m+1} *has a unique element and all the other sets are empty. Note that for the CCS composition over an infinite alphabet the* τ *accessibility relation cannot be defined by a determining sequence.*

(*5*) *In all the previous examples the determining sequences for accessibility relations do not use formulas* $\alpha \in ML$*. More general parallel composition operations in which the accessibility relations depend on the global state of a system can use such formulas. Consider an accessibility relation* R *defined by* $\langle a;\ \Diamond_b True;\ \beta(Z_1,\ X_1)\rangle$*, where* $\beta(Z_1, X_1) \triangleq \forall t.\ t \in Z_1 \wedge \exists t'.\ t' \in X_1$*. There is an* R *transition from* g *to* g' *iff for every* i *there is an* R_a *transition from* $g(i)$ *to* $g'(i)$ *in the structure* K_i *and there is at least one* j *such that* R_b *transition is possible from the state* $g(j)$ *in the structure* K_j*.*

3.3 Composition Theorem for Generalized products

THEOREM 21. **(Composition Theorem for the generalized product)** *Let* $\sigma = \langle A,\ V_c,\ B,\ V,\ \tau_{index},\ \rho\rangle$*, be a generalized product operator. For every formula* $\varphi \in ML(B, V)$ *there is a determining sequence* $\langle \alpha_1, \ldots, \alpha_n;\ \beta(X_1, \ldots, X_n)\rangle$ *where* α_i $(i = 1, \ldots, n)$ *are formulas in* $ML(A, V)$*. and* $\beta(X_1, \ldots, X_n)$ *in* $MSO(\tau_{index})$ *such that for every structure* $Ind = \langle I, \cdots\rangle$ *for* τ_{index} *and every family* K_i $(i \in I)$ *of* (A, V_c) *Kripke structures and a state* $g \in \Pi^\sigma_{i\in Ind} K_i$

$$\Pi^\sigma_{i \in Ind} K_i, g \models \varphi \text{ if and only if } Ind \models \beta(I^g_1, I^g_2, \ldots, I^g_n)\} \text{ where}$$

I^g_l $(l = 1, \ldots, n)$ *is the set* $\{i \in I\ :\ K_i, g(i) \models \alpha_l\}$*. Moreover, the formulas* α_i $(i = 1, \ldots, n)$ *and* β *are computable from* φ *and* σ*.*

Notes.

(1) Any modal formula φ defines a unary predicate $\{s\ :\ K, s \models \varphi\}$ over every Kripke structure K. The composition theorem can be rephrased as follows. For every generalized product operator σ and every formula φ corresponds a determining sequence γ such that φ and γ defines the same unary predicate for any σ-product $\Pi^\sigma_{i\in Ind} K_i$. Moreover, γ is computable from φ and σ.

(2) Let Ind be a fixed finite structure for τ_{index}, with the universe $\{1, \ldots n\}$. The composition theorem implies that for every generalized product operator σ and every formula φ there is a finite set M and modal formulas φ^j_i $(i \in M,\ j = 1, \ldots, n)$ such that for every sequence $K_1, \ldots, K_n$ of Kripke structures

$$\Pi^\sigma_{i=1,\ldots,n} K_i, \langle s_1, \ldots, s_n\rangle \models \varphi$$

if and only if for some $i \in M$ and all $j = 1, \ldots, n$

$$K_j, s_j \models \varphi^j_i.$$

Moreover, the formulas φ^j_i are computable from φ and σ and Ind.

As a consequence of Theorem 21 we obtain

COROLLARY 22. *There is an algorithm that solves the following decision problems in time $O(g(|\varphi|, n) \times (|K_1| + |K_2| + \cdots + |K_n|))$:*

(*1*) *Determine whether a state s of synchronous (respectively asynchronous) product of $K_1, \ldots, K_n$ satisfies φ.*

(*2*) *Determine whether a state s of $K_1|K_2|\cdots|K_n$ satisfies φ, where $|$ is CCS parallel composition.*

In the rest of this subsection the proof of Theorem 21 is given. We prove the theorem by induction on φ. However, unlike the proof of Theorem 2, the most subtle step is for modalities.

For atomic φ the assertion immediately follows from the definition of the product.

For disjunction inductively assume that the theorem holds for ψ and θ. Let $\langle \alpha_1, \ldots, \alpha_n;\ \beta(X_1, \ldots, X_n)\rangle$ and $\langle \alpha'_1, \ldots, \alpha'_m;\ \beta'(X_1, \ldots, X_m)\rangle$ be sequences correlated with ψ and θ. It is easy to check that the theorem holds when we correlate $\langle \alpha_1, \ldots, \alpha_n,\ \alpha'_1, \ldots, \alpha'_m;\ \beta(X_1, \ldots, X_n) \vee \beta'(X_{n+1}, \ldots, X_{n+m})\rangle$ with $\theta \vee \psi$.

For negation inductively assume that the theorem holds for ψ. Moreover suppose that $\langle \alpha_1, \ldots, \alpha_n;\ \beta(X_1, \ldots, X_n)\rangle$ is a sequence correlated with ψ. It is easy to check that the theorem holds when we correlate $\langle \alpha_1, \ldots, \alpha_n;\ \neg\beta(X_1, \ldots, X_n)\rangle$ with $\neg\psi$.

We say that $\langle \alpha_1, \ldots, \alpha_n;\ \beta(X_1, \ldots, X_n)\rangle$ and $\langle \alpha'_1, \ldots, \alpha'_m;\ \beta'(X_1, \ldots, X_m)\rangle$ are *equivalent* iff they define the same relations over every family K_i ($i \in Ind$) of Kripke structures

We need the following standard lemma.

LEMMA 23. *For every sequence $\langle \alpha_1, \ldots, \alpha_n;\ \beta(X_1, \ldots, X_n)\rangle$ there exists an equivalent $\langle \alpha'_1, \ldots, \alpha'_m;\ \beta'(X_1, \ldots, X_m)\rangle$ such that*

(*1*) $\bigvee \alpha'_i$ *is valid.*

(*2*) $\alpha'_i \wedge \alpha'_j$ *is unsatisfiable for $i \neq j$.*

PROOF. For $h \subseteq \{1, \ldots, n\}$ define

$$\alpha'_h \triangleq \bigwedge_{i \in h} \alpha_i \wedge \bigwedge_{i \notin h} \neg\alpha_i$$

It is clear that $\bigvee \alpha'_h$ is valid and $\alpha'_h \wedge \alpha'_g$ is unsatisfiable for $h \neq g$. Moreover, α_i is equivalent to $\bigvee_{i \in h} \alpha'_h$.

It is easy to check that $\langle \ldots, \alpha'_h, \ldots;\ \beta'(\ldots, X_h, \ldots)\rangle$ where h ranges over the subsets of $\{1, \ldots, n\}$ and $\beta'(\ldots, X_h, \ldots)$ is defined as

$$\beta'(\ldots, X_h, \ldots) \triangleq \exists X_1 \ldots \exists X_n. \bigwedge_{i=1}^{n} \big(\forall t. X_i(t) \leftrightarrow \bigvee_{i \in h} X_h(t)\big) \wedge \beta(X_1, \ldots, X_n)$$

is equivalent to $\langle \alpha_1, \ldots, \alpha_n;\ \beta(X_1, \ldots, X_n)\rangle$. $\square$

Now let us proceed with the inductive step for modalities. Assume that an accessibility relation for R_a is defined by a sequence

$$\langle a_1, \ldots, a_m;\ \alpha_1, \ldots \alpha_n;\ \beta(Z_1, \ldots, Z_m, X_1, \ldots, X_n)\rangle.$$

Inductively assume that the theorem holds for ψ. Moreover, let $\langle \gamma_1, \ldots, \gamma_l;\ \delta(Y_1, \ldots, Y_l)\rangle$ be a sequence correlated with ψ. By Lemma 23 we can assume that $\bigvee \gamma_i$ is valid and $\gamma_i \wedge \gamma_j$ is unsatisfiable for $i \neq j$.

We are going to construct a determining sequence for $\Diamond_a \psi$.

First, define $\theta_{k,r}$ (for $k = 1, \dots, m$ and $r = 1, \dots, l$) as

$$\theta_{k,r} \triangleq \Diamond_{a_k} \gamma_r$$

We correlate with $\Diamond_a \psi$ the sequence

$$\langle \alpha_1, \dots \alpha_n, \gamma_1, \dots, \gamma_l, \theta_{1,1}, \dots, \theta_{m,l};\ H(X_1, \dots, X_n, V_1, \dots V_l, V_{1,1}, \dots, V_{m,l}) \rangle, \quad (5)$$

where H says the following:

(A). There are disjoint sets $U_{k,r}$ ($k = 1, \dots, m$ and $r = 1, \dots, l$) such that $U_{k,r} \subseteq V_{k,r}$ and

(B). there are non-empty sets $Z_1, \dots, Z_m$ such that for $k = 1, \dots, m$

$$Z_k = \bigcup_r U_{k,r}$$

and

$$Ind \models \beta(Z_1, \dots, Z_m, X_1, \dots, X_n)$$

and

(C). there are $Y_1, \dots, Y_l$ such that for $r = 1, \dots, l$

$$Y_r = \big(\bigcup_k U_{k,r}\big) \cup \big(V_r \setminus \bigcup_{k,i} U_{k,i}\big)$$

and

$$Ind \models \delta(Y_1, \dots, Y_k).$$

Let us show the correctness of our construction. First, assume that g belongs to the predicate defined by the sequence (5). We are going to show that $\Pi^{\sigma}_{i \in Ind} K_i, g \models \Diamond_a \psi$. It is sufficient to define g' such that $g R_a g'$ and $\Pi^{\sigma}_{i \in Ind} K_i, g' \models \psi$. Take sets $U_{k,r}$, Z_k and Y_r which satisfy (A), (B) and (C). From (A) it follows that $j \in V_{k,r}$ for $j \in U_{k,r}$. Hence, $K_j, g(j) \models \Diamond_{a_k} \gamma_r$ (for $j \in U_{k,r}$). Therefore, there is $s_j \in K_j$ such that $K_j, s_j \models \gamma_r$ and $g(j) R_{a_k} s_j$. Define g' as follows

$$g'(j) = \begin{cases} s_j & \text{if } j \in U_{k,r} \\ g(j) & \text{otherwise} \end{cases}$$

Note that $g'(j)$ is well-defined because $U_{k,r}$ are disjoint. From the definition of g' and (B) it follows that $g(j) R_{a_k} g'(j)$ for $j \in Z_k$ and $g(j) = g'(j)$ for $j \notin \cup Z_k$. Therefore, from (A) and (B) it follows that $g R_a g'$. Observe that $K_j, g'(j) \models \gamma_i$ iff either $j \in U_{k,i}$ or $j \in V_i \setminus \bigcup U_{k,r}$ (in this case $g(j) = g'(j)$). Therefore, by (C) and the inductive assumption for ψ we obtain that $\Pi^{\sigma}_{i \in Ind} K_i, g' \models \psi$. Hence, $\Pi^{\sigma}_{i \in Ind} K_i, g \models \Diamond_a \psi$. This completes the first part of the proof.

Now let us show that if $\Pi^{\sigma}_{i \in Ind} K_i, g \models \Diamond_a \psi$ then g belongs to the predicate defined by the sequence (5). Let g' be such that $g R_a g'$ and $\Pi^{\sigma}_{i \in Ind} K_i, g' \models \psi$.

From the definition of R_a it follows that there are disjoint $Z_1, \dots, Z_l$ such that

$$g(j) R_{a_i} g'(j) \text{ for } j \in Z_i$$

and

$$g'(j) = g(j) \text{ for } j \notin \bigcup Z_i$$

$$Ind \models \beta(Z_1, \ldots, Z_m, X_1, \ldots, X_n)$$

where $X_k = \{i \in I \ : \ K_i, g'(i) \models \alpha_k\}$.

Since $\Pi^{\sigma}_{i \in Ind} K_i, g' \models \psi$, we have that

$$Ind \models \delta(Y_1, \ldots, Y_l)$$

where $Y_i = \{j \ : \ K_j, g'(j) \models \gamma_i\}$. Note that $Y_1, \ldots, Y_l$ are disjoint because $\gamma_i \wedge \gamma_{i'}$ are unsatisfiable for $i \neq i'$. Hence, $U_{k,r} \triangleq Z_k \cap Y_r$ are disjoint. We leave for the reader to verify that (A), (B) and (C) hold.

3.4 Composition Theorem Fails for Logics with reachability modality

In section 2.4 we proved that the composition theorem over n-products ("products without communication") and the modal logic extended by the reachability modality **EF** holds. (Recall that $\mathbf{EF}p$ holds at s iff a node with property p is reachable from s.) However, the composition theorem fails over n-products and the modal logic extended by the modality **EG**. Here we show that the composition theorem fails over the generalized products and any logic which has the reachability modality **EF**. More precisely we will show that the composition theorem fails even over the synchronous product (a very simple and basic instance of the generalized product) for any logic which can express reachability **EF**.

The synchronous product of frames is an important instance of generalized product. It was defined in Example 12 in subsection 3.1. Recall that it is defined as follows. Let $A = \{a_1, \ldots, a_m\}$ be a set (of action). Given a family of A-frames $\mathcal{F}_1 = \langle S_1, \ R^1_{a_1}, R^1_{a_2}, \ldots, R^1_{a_m}\rangle, \ldots, \mathcal{F}_n = \langle S_n, \ R^n_{a_1}, R^n_{a_2}, \ldots, R^n_{a_m}\rangle, \ldots$, their synchronous product is defined to be the A-frame $\langle S_1 \times \cdots \times S_n \times \cdots, \ R_{a_1}, \ldots, R_{a_m}\rangle$, where R_{a_i} $(i = 1, \ldots, m)$ is the following relation on $S_1 \times \cdots \times S_n \times \cdots$:

$$\langle s_1, \ldots, s_n, \ldots\rangle R_{a_i} \langle s'_1, \ldots, s'_n, \ldots\rangle \text{ iff } s_n R^n_{a_i} s'_n \text{ for all } n.$$

We denote by $\mathcal{F}_1 \bigotimes_{sync} \mathcal{F}_2$ the synchronous product of two frames $\mathcal{F}_1$ and $\mathcal{F}_2$. The synchronous products of Kripke structure define the accessibility relations like the synchronous product of the underlying frames, and provide an interpretation by (arbitrary) determining sequences for unary predicates.

Consider a frame $D_n = \langle S_n, \ R_a, \ R_b\rangle$ over two actions $\{a, \ b\}$ defined as follows:

States:. the universe S_n is $\{0, 1, 2, 3, \ldots, n+1\}$;

Accessibility relations:. R_a is interpreted as $R_a = \{\langle i, \ i+1\rangle \ : \ i = 0, \ldots, n-1\}$ and R_b contains only one pair $\langle n, n+1\rangle$.

We identify the frame D_n with the Kripke structure over D_n with no unary predicates.

The formula $\mathbf{EF}\Diamond_b$True holds at a state s in a Kripke structure if it is possible to reach from s a state where b transition is possible. The following lemma is immediate.

LEMMA 24. *The state $\langle 0, 0\rangle$ of $D_n \bigotimes_{sync} D_m$ satisfies $\mathbf{EF}\Diamond_b$True if and only if $n = m$.*

By the same argument as in the proof of Theorem 11, one can derive from Lemma 24

THEOREM 25 (COMPOSITION THEOREM FAILS FOR LOGICS WITH REACHABILITY). *Let L be any logic which can express $\mathbf{EF}\Diamond_b$True. There is no finite set I and a sequence of L-formulas $\phi^i_1, \ \phi^i_2$ $(i \in I)$ such that*

$$K_1 \bigotimes_{sync} K_2, \langle s_1, s_2\rangle \models \mathbf{EF}\Diamond_b \mathit{True} \text{ if and only if}$$

$$K_1, s_1 \models \phi_1^i \text{ and } K_2, s_2 \models \phi_2^i \text{ for some } i \in I.$$

4. PARAMETERIZED SYSTEMS

Here we provide applications of composition theorem for parametric model checking.

Many protocols are specified by a number of instances of identical processes. Sometimes there is an infinite number of possible instances. Each instance can be represented as a finite product of identical (or similar) Kripke structures. One is usually interested to show that all the instances have a certain property. Here we suggest a formalization of parameterized systems of processes as the generalized power. This formalization encompasses many constructions considered in the literature [German and Sistla 1992; Emerson and Namjoshi 1998]. Let $\sigma = \langle A,\ V_c,\ B,\ V,\ \tau_{\text{index}},\ \rho\rangle$, be a generalized product operator. Let $Ind = \langle I, \cdots\rangle$ be a structure for τ_{index} and let K be an (A, V_c) Kripke structures. We denote by $power(K, Ind, \sigma)$ the generalized product $\Pi^{\sigma}_{i\in Ind} K_i$ where all K_i are isomorphic to K.

For a class C of τ_{index} structures and K and σ as above we denote by $power(K, C, \sigma)$ the class $\{power(K, Ind, \sigma)\ :\ Ind \in C\}$ of structures.

Example 26 Token Ring Protocol. *Our formalization of the token ring protocol follows the presentation in [German and Sistla 1992]. The processes are arranged on a ring. Initially one process has a token that permits it to enter its critical region. The processes circulate the token around a ring network. All processes have the same behavior as described by the Kripke structure K in Figure 3.*

The critical region of K consists of the state C. A process in state N can enter its waiting state W by its internal transition I. Then, it waits to receive the token from its left neighbour before it can enter its critical region. The state T is used by a process that has the token, but is not in its critical region. It permits the token to be circulated by processes without entering the critical region.

The instance S_n of this protocol over the ring of size n can be described as follows. The states are all the functions from $\{0, \ldots, n-1\}$ to the states of K (see Figure 3). The structure S_n is a Kripke structure for one accessibility relation. There is a transition

$$\langle s_0, \ldots, s_{n-1}\rangle \rightarrow \langle s'_0, \ldots, s'_{n-1}\rangle$$

if either (1) exactly one component j executes the internal transition I from s_j to s'_j and all the other components do not move, i.e. $s_i = s'_i$ for $i \neq j$ or (2) there is j such that j-th component can move from s_j to s'_j by an R transition and its right neighbour ($r = j + 1 \bmod n$) can move from s_r to s'_r by an L transition, and all the other components do not move.

It should be clear how to formalize this structure S_n as a generalized power of K over the directed circle $Circ_n$ of size n considered as the structure for the binary relation $Edge(\ ,\ )$. For example, Condition (1) is formalized by the following determining sequence for an accessibility relation:

$$\langle I;\ ;\exists! t.\ t \in Z_1)\rangle.$$

Condition (2) is formalized by the following determining sequence for an accessibility

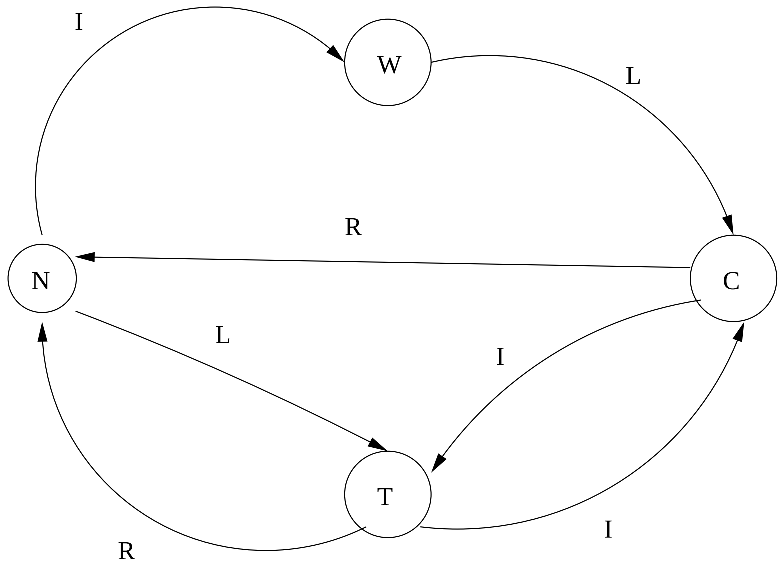

Fig. 3.

relation:

$\langle L, R;\ ; (\exists! t_1\ t_1 \in Z_1) \wedge (\exists! t_2\ t_2 \in Z_2) \wedge (\exists t_1 \exists t_2\ (t_1 \in Z_1) \wedge (t_2 \in Z_2) \wedge Edge(t_1, t_2))\rangle$.

Both these sequences do not contain any α formula. The accessibility relation for the token ring protocol is defined as the disjunction $\langle I, L, R;\ ;: \beta(Z_1, Z_2, Z_3)\rangle$ of these two sequences, where β is

$$\begin{gathered}(\exists! t_1\ t_1 \in Z_1) \wedge \neg(\exists t\ t \in Z_2) \wedge \neg(\exists t\ t \in Z_3) \\ \vee \\ \neg \exists t_1\ t_1 \in Z_1 \wedge \exists! t_2\ t_2 \in Z_2 \wedge \exists! t_3\ t_3 \in Z_3 \wedge (\exists t_2 \exists t_3\ t_2 \in Z_2 \wedge t_3 \in Z_3 \wedge Edge(t_2, t_3)).\end{gathered}$$

One can also define on S_n unary predicates such as $C^{\leq 1}$ - at most one component is in the state C, or $T^{=1}$ - there is exactly one component with the token.

For a formula φ one is usually interested in whether φ holds on S_n for every n. The most relevant property of the token ring protocol is the mutual exclusion (in all reachable states $C^{\leq 1}$ holds). Unfortunately, multi-modal logic is too weak to express the mutual exclusion. However, the invariant $(T^{=1} \wedge C^{\leq 1}) \rightarrow \Box(T^{=1} \wedge C^{\leq 1})$ implies the mutual exclusion.

The Parametric Model Checking Problem (PMCP) over a class C of structures for a generalized product σ and for a logic L can be formalized as follows:

> For a formula $\varphi \in L$ and a finite state Kripke structure K decide whether φ is valid over $power(K, C, \sigma)$.

As a consequence of Theorem 21 we obtain

COROLLARY 27. *If the monadic second order theory of C is decidable then PMCP over C for modal logic is decidable in time $O(g(|\varphi|) \times |K|)$ where g is a recursive function.*

Proof. (Sketch) First, observe that every determining sequence for a unary predicate is equivalent to a sequence of the form $\langle \alpha_1, \ldots, \alpha_n, \beta(X_1, \ldots X_n)\rangle$ where α_i and α_j are inconsistent (i.e. $\alpha_i \wedge \alpha_j$ is unsatisfiable) for $i \neq j$.

The algorithm proceeds as follows. Given a formula φ.

(1) Construct a determining sequence $\langle \alpha_1, \ldots, \alpha_n, \beta(X_1, \ldots X_n)\rangle$ associated with φ as in Theorem 21. Moreover, we can assume that α_i and α_j are inconsistent for $i \neq j$.

(2) Find $J = \{j \ : \ \alpha_j \text{ is satisfiable in } K\}$.

(3) Check if the formula

$$\forall X_1 \ldots \forall X_n (\bigwedge_{j \notin J} \mathit{Empty}(X_j) \wedge \bigwedge_{j_1 \neq j_2} \mathit{Empty}(X_{j_1} \cap X_{j_2})) \rightarrow \beta(X_1, \ldots, X_n) \quad (6)$$

holds over C (here $\mathit{Empty}(X)$ abbreviates $\forall t. t \notin X$).

Note that (6) holds over C iff the formula φ holds over $\mathit{power}(K, C, \sigma)$. Observe that the complexity of the first and the third steps are independent from K. The complexity of the second step is linear in $|K|$, by Theorem 3. These observations imply the complexity bound stated in the corollary. ☐

The monadic second order theory of the class of circles is decidable (it can be easily interpreted in the monadic second order theory of finite linear orders). Hence, we deduce that PMCP for modal logic is decidable for the class of ring protocols.

5. CONCLUSION

Composition Theorems are tools which reduce sentences about some compound structures to sentences about their parts. A seminal example of such a result is the Feferman-Vaught Theorem [Feferman and Vaught 1959] which reduces the first-order theory of generalised products to the first order theory of its factors. Composition theorems for monadic second-order logic and generalized sums were developed by Shelah [Shelah 1975]. The technique was used in [Gurevich 1979; Gurevich and Shelah 1979; 1983; 1985; Hafer and Thomas 1987; Moller and Rabinovich 1999; 2003; Courcelle et al. 2000; Makowsky 2004] and is outlined in survey expositions by Gurevich [Gurevich 1985] and Thomas [Thomas 1997].

The aim of our work was to explore the applicability of this approach in the area of verification. We obtained a positive result - the composition theorem is realizable when the specification language L_{spec} is the multi-modal logic and the set of operations OP consists of a wide variety of the generalized product ("parallel composition") operators, and a negative result - if L_{spec} can express "there is a path such that all the nodes of the path have a property p", then the composition theorem fails for very simple parallel operators.

The generalized product of Kripke structures suggested here is a proper instance of Feferman-Vaught generalized product construct [Feferman and Vaught 1959]. Many minor modifications of the generalized product construct of Kripke structures are not appropriate. They lead to one of the following situations: (1) Kripke structures $S_1, \ldots, S_n$ might be bisimulation equivalent to $S'_1, \ldots, S'_n$, but the product of S_i is not bisimulation equivalent to the product of S'_i. (2) The composition theorem for multi-modal logics fails over these more general products.

Our composition theorem is a strong inductive assertion. When such an assertion is stated correctly its proof is easy and proceeds by the standard arguments developed by Feferman-Vaught [Feferman and Vaught 1959].

The negative results show that (1) the composition theorem fails even for very simple product operations (which correspond to the parallel composition without communication) when the multi-modal logic is replaced by any logic which can express **EG**p; (2) the composition theorem fails over the synchronous product for any logic which has the reachability modality. Though the proofs of these results are simple, the results are important because they provide a very sharp bound on the limitations of compositional methods. The second result was recently complemented by a theorem of Wöhrle and Thomas [Wöhrle and Thomas 2004], which shows that "semi-finite synchronization" does not preserve the decidability of $FO(R)$ - the first-order logic extended by the reachability modality (i.e., there are Kripke structures with decidable model checking problem for $FO(R)$, but the model checking problem for $FO(R)$ over their semi-finite synchronization product is undecidable).

It is important to emphasize that the composition theorem for a set of operations OP and a specification language L_{spec} is much stronger than other notions of compositionality considered in the literature on verification (e.g., see [Owicki and Gries 1976; Lamport 1980; de Roever 1985; Stirling 1988; Zwiers 1989; Apt and Olderog 1991; de Roever 1997; Trakhtenbrot 1997]). In such compositional frameworks (see [de Roever 1997]) it is required that for every operator $op^P \in OP$‘ there should exist an operator op^S in the specification language such that

(1) Whenever P_i satisfy specifications φ_i for $i = 1, \ldots n$, one also has for every n-ary operator $op^P \in OP$ that $op^P(P_1, \ldots P_n)$ satisfies $op^S(\varphi_1, \ldots, \varphi_n)$.

(2) Whenever $op^P(P_1, \ldots P_n)$ satisfies φ, there exist specifications φ_i for P_i such that P_i satisfies φ_i for $i = 1, \ldots n$ and $op^S(\varphi_1, \ldots, \varphi_n) \rightarrow \varphi$ is valid.

It is easy to see that the composition theorem implies the above properties, but it does not follow from them.

Often arguments which show that there is no compositional proof system for L_{spec} and OP exploit that the specification language L_{spec} is too weak (with respect to OP). One usually shows that two programs P_1 and P_2 satisfy the same specifications in L_{spec}, but there is a context $op[\ , Q]$ and a formula φ such that $op(P_1, Q)$ satisfies φ, while $op(P_2, Q)$ does not satisfy φ. In such situations in order to gain a compositional proof system L_{spec} should be replaced by a more expressive language. Our negative result shows that the increase in the expressive power of L_{spec} cannot help to obtain the composition theorem.

Acknowledgements

I would like to thank Colin Stirling for his valuable suggestions and the anonymous referees for their detailed and insightful comments which helped us to clarify some unclear points in the first version.

REFERENCES

Apt, K. R. and Olderog, E. R. 1991. *Verification of sequential and concurrent programs.* Springer-Verlag.

CLARKE, E. M. AND EMERSON, E. A. 1981. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*. Lecture Notes in Computer Science, vol. 131. Springer, 52–71.

COURCELLE, B., MAKOWSKY, J. A., AND ROTICS, U. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst. 33,* 2, 125–150.

DE ROEVER, W. P. 1985. The quest for compositionality - a survey of assertion-based proof systems for concurrent programs. In *Proc. of the IFIP Working Conference The role of abstract models in computer science.* North-Holland.

DE ROEVER, W. P. 1997. The need for compositional proof systems: A survey. In *COMPOS*. Lecture Notes in Computer Science, vol. 1536. Springer, 1–22.

EMERSON, E. A. AND NAMJOSHI, K. S. 1996. Automatic verification of parameterized synchronous systems (extended abstract). In *CAV*. Lecture Notes in Computer Science, vol. 1102. Springer, 87–98.

EMERSON, E. A. AND NAMJOSHI, K. S. 1998. On model checking for non-deterministic infinite-state systems. In *LICS*. 70–80.

ESPARZA, J., FINKEL, A., AND MAYR, R. 1999. On the verification of broadcast protocols. In *LICS*. 352–359.

FEFERMAN, S. AND VAUGHT, R. 1959. The first-order properties of products of algebraic systems. *Fundamenta Mathematica 47*, 57–103.

GABBAY, D. M. AND SHEHTMAN, V. B. 1998. Products of modal logics, part 1. *Logic Journal of the IGPL 6,* 1, 73–146.

GABBAY, D. M. AND SHEHTMAN, V. B. 1999. Flow products of modal logics.

GERMAN, S. M. AND SISTLA, A. P. 1992. Reasoning about systems with many processes. *J. ACM 39,* 3, 675–735.

GUREVICH, Y. 1979. Modest theory of short chains. i. *J. Symb. Log. 44,* 4, 481–490.

GUREVICH, Y. 1985. *Monadic second-order theories, In Model-Theoretical Logics (ed. J. Barwise and S. Feferman)*. Springer-Verlag, 479–506.

GUREVICH, Y. AND SHELAH, S. 1979. Modest theory of short chains. ii. *J. Symb. Log. 44,* 4, 491–502.

GUREVICH, Y. AND SHELAH, S. 1983. Rabin's uniformization problem. *J. Symb. Log. 48,* 4, 1105–1119.

GUREVICH, Y. AND SHELAH, S. 1985. The decision problem for branching time logic. *J. Symb. Log. 50,* 3, 668–681.

HAFER, T. AND THOMAS, W. 1987. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *ICALP*. Lecture Notes in Computer Science, vol. 267. Springer, 269–279.

HENNESSY, M. AND MILNER, R. 1985. Algebraic laws for nondeterminism and concurrency. *J. ACM 32,* 1, 137–161.

HIRSCH, R., HODKINSON, I. M., AND KURUCZ, Á. 2002. On modal logics between K x K x K and S5 x S5 x S5. *J. Symb. Log. 67,* 1, 221–234.

LADNER, R. E. 1977. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput. 6,* 3, 467–480.

LAMPORT, L. 1980. The 'Hoare logic' of concurrent programs. *Acta Inf. 14*, 21–37.

MAKOWSKY, J. A. 2004. Algorithmic aspects of the Feferman-Vaught theorem. *Annals of Pure and Applied Logic 126(1-3)*, 159–213.

MILNER, R. 1989. *Communication and Concurrency.* Prentice-Hall.

MOLLER, F. AND RABINOVICH, A. 1999. On the expressive power of CTL. In *LICS*. 360–369.

MOLLER, F. AND RABINOVICH, A. 2003. Counting on CTL^*: on the expressive power of monadic path logic. *Inf. Comput. 184,* 1, 147–159.

OWICKI, S. S. AND GRIES, D. 1976. An axiomatic proof technique for parallel programs i. *Acta Inf. 6*, 319–340.

SHELAH, S. 1975. The monadic theory of order. *Annals of Mathematics 102*, 379–419.

STIRLING, C. 1988. A generalization of Owicki-Gries's hoare logic for a concurrent while language. *Theor. Comput. Sci. 58*, 347–359.

THOMAS, W. 1997. Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In *Structures in Logic and Computer Science*. Lecture Notes in Computer Science, vol. 1261. Springer, 118–143.

TRAKHTENBROT, B. A. 1997. On the power of compositional proofs for nets: Relationships between completeness and modularity. *Fundam. Inform. 30,* 1, 83–95.

VAN BENTHEM, J. 1976. Modal correspondence theory. Ph.D. thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam.

WÖHRLE, S. AND THOMAS, W. 2004. Model checking synchronized products of infinite transition systems. In *LICS*. 2–11.

ZWIERS, J. 1989. *Compositionality, Concurrency and Partial Correctness - Proof Theories for Networks of Processes, and Their Relationship.* Springer.

Received August 2003; revised July 2004; accepted November 2004