View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

Graphical VERSUS logical specifications

Gérard Boudol, Kim Guldstrand Larsen

To cite this version:

Gérard Boudol, Kim Guldstrand Larsen. Graphical VERSUS logical specifications. [Research Report] RR-1104, INRIA. 1989. inria-00075455

HAL Id: inria-00075455

https://hal.inria.fr/inria-00075455

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1104

Programme 1
Programmation, Calcul Symbolique
et Intelligence Artificielle

GRAPHICAL VERSUS LOGICAL SPECIFICATIONS

Gérard BOUDOL
Kim G. LARSEN

Octobre 1989

Graphical versus Logical Specifications

Specifications Graphiques versus Logiques

Gérard Boudol
INRIA Sophia-Antipolis
06565-VALBONNE FRANCE

Kim G. Larsen
Ålborg University
9000-ÅLBORG DENMARK

Résumé.

Nous étudions les relations qui existent entre deux formalismes de spécification et les méthodes de vérification associées pour les systèmes parallèles. Ces deux formalismes sont d'une part les "systèmes de transitions modaux" de Larsen et Thomsen – que nous appelons ici spécifications graphiques –, et la logique de Hennessy et Milner de l'autre. Nous montrons que toute spécification graphique peut être exprimée par une spécification logique, qui a les mêmes modèles. Inversement, nous donnons une caractérisation des formules qui sont graphiquement représentables.

Abstract.

This paper studies the relations between two specification formalisms and verification methods for concurrent systems, namely Larsen-Thomsen's modal transition systems – here called graphical specifications – and Hennessy-Milner Logic. We show that any graphical specification may be expressed by a logical specification having the same models. Conversely, we give a characterization of the formulae that are graphically representable.

PAPIER RECUPERÉ ET RECYCLÉ

Graphical versus Logical Specifications

Gérard Boudol
INRIA Sophia-Antipolis
06565-VALBONNE FRANCE

Kim G. Larsen
Ålborg University
9000-ÅLBORG DENMARK

Abstract.

This paper studies the relations between two specification formalisms and verification methods for concurrent systems, namely Larsen-Thomsen's modal transition systems – here called graphical specifications – and Hennessy-Milner Logic. We show that any graphical specification may be expressed by a logical specification having the same models. Conversely, we give a characterization of the formulae that are graphically representable.

1. Introduction.

The purpose of this paper is to investigate the relationships between two approaches to the verification of concurrent systems: the logical approach, and what we call the graphical approach.

Following the logical approach, a specification is a set of properties, that is a set of formulae of some logic, and verification is a model-checking activity. We denote "the process P is a model of the formula ϕ" by

$$P \models \phi$$

The model we shall use for concurrent processes is that of labelled transition systems. This is a common model for many existing verification tools, *cf.* for instance [1,2,3,4,11]. The properties that one states about these systems are expressed in Hennessy-Milner Logic [6].

In the graphical approach, processes and specifications are systems of the same kind – namely labelled transition systems –, and the verification amounts to compare such systems. We call this approach graphical because a transition system is just a special case of graph. Indeed a process or a specification is best understood by means of drawings, which moreover can be directly represented on a screen and interpreted by a computer, using a graphical editor (*cf.* [2]).

In many cases, the comparison of a process to a graphical specification is based on the notion of bisimulation: the process satisfies the specification if they are equivalent, up to some bisimulation depending on a particular observation criterion (*cf.* [11]). The resulting verification method is related to the logical one since two systems are bisimilar if and only if they satisfy the same formulae. This is the standard Hennessy-Milner Theorem [6]:

$$P \sim S \;\Leftrightarrow\; \forall \phi.\, P \models \phi \;\Leftrightarrow\; S \models \phi$$

In comparing these two approaches to specification and verification (for a survey, see [13]), we may use several criteria:

• *expressivity*: the specification formalism should be powerful enough to express all the properties of a given process. In other words, it should be possible to completely specify any labelled transition system, up to bisimulation.

• *modularity*: processes are often made out of several components, and it may be the case that some properties satisfied by the components are enough to ensure a property of the global system. Then we would expect a "reusability of proofs": if one replaces a component by another satisfying the same sub-specification, then the whole specification should remain valid.

• *refinement*: one should have the ability to deal with partial specifications, requiring more and more properties about a system, up to its complete specification.

Applying these criteria to the logical and the graphical (i.e. bisimulation) frameworks, we get the following situation:

	logical	*graphical*
expressivity	*yes*	*trivial*
modularity	*yes?*	*yes*
refinement	*trivial*	*no*

As one can see, the logical and graphical frameworks offer complementary advantages: on the graphical side, expressivity is trivial since a process is a specification of itself. Modularity is usually granted by the fact that bisimulations are compatible with (most) process constructors; for instance, the well-known weak bisimulation is a congruence with respect to the static operators of CCS (parallel composition, restriction, renaming).

On the logical side, expressivity is achieved if we allow possibly infinite sets of formulae as logical specifications – this is the content of the Hennessy-Milner Theorem. In some cases one can even characterize a system by a single formula (*cf.* [5,13]). The point of modularity is more dubious: for instance in [5] a logic is presented which incorporates some process constructors. However, according to [14] these operators are not logical (truth functional), and verification in such an extended logic involves graphical manipulations (quotient w.r.t. bisimulation).

Regarding refinement, the two approaches are strikingly different: in the logical setting there is a natural refinement preorder on specifications (= sets of formulae), namely the inclusion of their classes of models. Restricted to formulae this preorder is:

$$\phi \leq \psi \;\Leftrightarrow\; \forall P.\; P \models \phi \;\Rightarrow\; P \models \psi$$

From a practical point of view, one refines a logical specification simply by adding new requirements – note that this may lead to inconsistencies. On the other hand, there is no gradualness in graphical verification. How could we remedy this deficiency, and find an image of the logical refinement preorder in a graphical setting?

Although this question was not his direct motivation, Milner showed in [12] (*cf.* also [14]) that the logical preorder corresponds to an observational preorder for a generalised notion of process, namely labelled transition systems equipped with a divergence predicate. The semantics of the logic has to be adapted to obtain this result.

A different model, explicitly intended to provide a notion of graphical specification supporting refinement, was recently introduced in [9]. In this model, a *graphical specification* (called there "modal transition system") is a labelled transition system together with a predicate on transitions. This predicate asserts that some transitions must exist in any model of the specification; on the

other hand, a model is allowed to perform only the transitions mentioned in the specification. Then one can define a refinement preorder between graphical systems,

$$S \sqsubseteq P$$

which can be read "the specification P is finer than S", or also "P is a model of the specification S" (for lack of space, we shall not give examples motivating and illustrating the use of graphical specifications and refinement; these may be found in [9,7,8]). The notion of graphical specification contains the usual notion of process: a process is a graphical specification where the "certainty" predicate is total; moreover the refinement preorder generalises strong bisimulation. Therefore this model for specifications is expressive with respect to the class of labelled transition systems. Moreover graphical specifications may be composed, and it has been shown in [9,7] that the refinement preorder is compatible with the usual process constructors; therefore this framework supports modular verifications.

The graphical approach seems to provide a well-behaved specification formalism; to confirm its adequacy, we want now to study its relations with the logical formalism. It is clear that we cannot capture "graphically" the whole logic, since any graphical specification is consistent. So, what is lost? Not so much, as we shall see: logic is the join-semilattice (with bottom) generated by the finite, acyclic graphical specifications.

Let us now present more precisely our results. First we will show that Hennessy-Milner Logic has a natural interpretation with respect to graphical specifications, and that their theorem extends easily to this setting: a graphical specification is a refinement of another one if and only if it satisfies more formulae, that is:

$$S \sqsubseteq P \;\Leftrightarrow\; \forall \phi.\, S \models \phi \;\Rightarrow\; P \models \phi$$

We will give a syntax for the finite, acyclic(†) graphical specifications – the corresponding set of terms is denoted by Θ. We will then show that each term $t \in \Theta$ may be represented by a characteristic formula $\chi(t)$, in the sense that:

$$\begin{aligned} t \sqsubseteq t' &\;\Leftrightarrow\; \chi(t') \leq \chi(t) \\ &\;\Leftrightarrow\; t' \models \chi(t) \end{aligned}$$

In other words: to verify that a system satisfies the specification t amounts to check that this system is a model of the formula $\chi(t)$ (or to check the validity of $\chi(t') \leq \chi(t)$, but this is in general quite complex).

Our main interest was in exploring the converse relation: to what extent can logical model-checking be achieved by means of graphical manipulations? We would like to know which kind of formula ϕ can be *graphically represented* by a specification $\theta(\phi)$ such that:

$$S \models \phi \;\Leftrightarrow\; \theta(\phi) \sqsubseteq S$$

We show that this holds exactly when ϕ is a (consistent) *prime* formula, that is irreducible with respect to disjunction – and in this case $\theta(\phi)$ may be described by a term of Θ. We can then summarize the content of the paper by saying that there is a Galois connection between the preordered set $(\Pi, \leq)$ of consistent prime formulae and the preordered set $(\Theta, \sqsubseteq)$ of finite acyclic graphical specifications. Moreover any formula is logically equivalent to a finite disjunction of prime formulae, therefore the logic can be regarded as generated by graphical specifications of Θ.

(†) this is slightly inaccurate: we allow as a constant the loosest specification, which is cyclic.

2. Graphical Specifications and Refinement.

2.1 Specifications.

A graphical specification, also called modal transition system in [9], is a labelled transition system equipped with a predicate on transitions. As a specification, such a system is intended to describe a whole class of possible realizations, rather than a particular process. Then the transition relation conveys the following meaning: it *allows* a model to perform the specified transitions, but the *existence* of these transitions is not necessarily required. On the other hand, if the specification does not mention a particular transition, then this transition is disallowed from any possible model. The remaining ingredient of graphical specifications, the predicate on transitions, asserts that some transitions are required: these transitions *must exist* in any model of the specification.

The formal definition is as follows: we assume given a finite set A of actions (finiteness is not a serious restriction, see the remarks below); then a *graphical specification system* S is a system $(Q, \vartheta, !)$ where

(i) Q is the set of states;

(ii) $\vartheta \subseteq Q \times A \times Q$ is the transition relation, which is *image finite*, i.e. for all $q \in Q$ and $a \in A$ the set $\{q' \mid (q, a, q') \in \vartheta\}$ is finite;

(iii) $!$ is a predicate on transitions, i.e. $! \subseteq \vartheta$.

A *graphical specification* is a pair (S, s) where $S = (Q, \vartheta, !)$ is a specification system and $s \in Q$ an initial state.

NOTATIONS and TERMINOLOGY: we shall use S, T, $P \ldots$ to range over specification systems, p, q, $s \ldots$ to range over states, and a, b, $c \ldots$ to denote actions. For any specification system $S = (Q, \vartheta, !)$, we shall denote $p \overset{a}{\underset{S}{\rightsquigarrow}} q$ (or simply $p \overset{a}{\rightsquigarrow} q$) the fact that $(p, a, q) \in \vartheta$. These are the *possible* transitions allowed by S, whereas the transitions satisfying $!$ are the *definite* transitions required by the specification. We shall use the notation $p \overset{a}{\underset{S}{\longrightarrow}} q$ (or simply $p \overset{a}{\longrightarrow} q$) for such definite transitions. The system S is a *process system* if every transition is required, that is $! = \vartheta$; in this case we simply denote S by (Q, ϑ). A *process* is a specification (S, s) where S is a process system.

The original definition ([9]) was given in terms of two kinds of transitions: the *may* transitions $p \overset{a}{\longrightarrow}_\diamond q$, corresponding to our $p \overset{a}{\rightsquigarrow} q$, and the *must* transitions $p \overset{a}{\longrightarrow}_\Box q$, related to the previous ones by a *consistency* assumption:

$$p \overset{a}{\longrightarrow}_\Box q \;\Rightarrow\; p \overset{a}{\longrightarrow}_\diamond q$$

This axiom asserts that if a state is required to perform the action a, then it is also allowed to do so. The "must" transitions correspond to our $!$ predicate.

Graphical specifications are best visualized by means of drawings, but for many purposes it is convenient to have a syntax. To relate the graphical and logical frameworks, we shall only need a very elementary syntax – richer ones, including fixpoints or parallel composition, may be found in [9,8]. The terms we shall deal with are given by the following grammar:

$$t ::= 0 \mid \omega \mid a.t \mid a!t \mid (t + t)$$

where a stands for any action. We use two prefixing constructs $a.t$ and $a!t$, corresponding respectively to the "may" and "must" transitions. The sum $t_0 + t_1$ is like the CCS one, and 0 is an "empty" specification – disallowing any action –, similar to CCS "nil". We shall denote the set of terms by Θ, and use t, u, $v \ldots$ to range over terms.

We now interpret these terms as graphical specifications. More specifically, each term t is a state of a graphical system $\mathcal{G}$ whose transitions are given by the following rules:

$$\forall a \in A \quad \omega \overset{a}{\rightsquigarrow} \omega$$

This is the "loosest" specification (denoted $\mathcal{U}$ in [9]): ω does not require anything, while allowing any possible move. The axioms for guarded terms are:

$$a.t \overset{a}{\rightsquigarrow} t \qquad\qquad a!t \overset{a}{\rightarrow} t$$

We recall that a transition $a!t \overset{a}{\rightarrow} t$ is both allowed (i.e. $a!t \overset{a}{\rightsquigarrow} t$) and required. Therefore the term $a.t$ allows initially an a action, while $a!t$ requires this action, and both initially disallow any other transition. The rules for sum are:

$$\frac{t_i \overset{a}{\rightsquigarrow} t}{t_0 + t_1 \overset{a}{\rightsquigarrow} t} \qquad\qquad \frac{t_i \overset{a}{\rightarrow} t}{t_0 + t_1 \overset{a}{\rightarrow} t} \qquad (i = 0, 1)$$

We shall freely use the notation $\sum_{i \in I} t_i$ for a finite summation (which is 0 if $I = \emptyset$). For instance, the term $\sum_{b \in B} b.\omega$, where B is a set of actions, represents a specification allowing only actions in B to occur initially. This term will be denoted σ_B.

REMARK. To deal with an infinite alphabet of actions, we should introduce a constant $\sigma_B = \sum_{b \in B} b.\omega$ for any cofinite set B of actions (that is $B \subseteq A$ and $A - B$ is finite).

2.2 Refinement.

In this section we formalize what it means for a system to satisfy, or be a realization, of a specification. Intuitively, a specification is satisfied by a system if

(i) whenever the specification requires a transition, then a corresponding transition exists for sure in the system,

and

(ii) if a transition is possible in the system then it is allowed by the specification.

In fact we shall define a relation $(S, s) \sqsubseteq (P, p)$ between specifications, which should be read "the specification (S, s) is satisfied by (P, p)", or "(P, p) is a refinement of (S, s)". If (P, p) is a process, then we could say that (P, p) is a model, or an implementation of (S, s). We first define a notion of refinement between states of a given graphical system. Given a specification system $S = (Q, \vartheta, !)$, a *refinement* over S is a relation $R \subseteq Q \times Q$ satisfying:

if $p\,R\,q$ then

(i) $p \overset{a}{\rightarrow} p' \;\Rightarrow\; \exists q'\; q \overset{a}{\rightarrow} q' \;\&\; p'\,R\,q'$

(ii) $q \overset{a}{\rightsquigarrow} q' \;\Rightarrow\; \exists p'\; p \overset{a}{\rightsquigarrow} p' \;\&\; p'\,R\,q'$

As usual, we are interested in the largest such relation, that is:

$$p \sqsubseteq q \;\Leftrightarrow_{\text{def}}\; \exists R \text{ refinement} \quad p\,R\,q$$

and the following holds (*cf.* [9,7]):

LEMMA. *The relation $\sqsubseteq$ is a refinement and a preorder.*

This relation is extended to specifications (S,s) as follows: given two specification systems $S = (Q,\vartheta,!)$ and $P = (Q',\vartheta',!')$, we denote by $S \uplus P$ their disjoint union – with an obvious definition, the set of states being $(\{0\} \times Q) \cup (\{1\} \times Q')$ –, and we shall say that $(S,s) \sqsubseteq (P,p)$ if and only if $(0,s) \sqsubseteq (1,p)$ in the system $S \uplus P$. We still say that (P,p) *refines* or *satisfies* the specification (S,s) when $(S,s) \sqsubseteq (P,p)$ holds. For instance $(\mathcal{G},\omega) \sqsubseteq (S,s)$ for any specification (S,s). The equivalence associated with the refinement preorder is denoted $\simeq$, thus:

$$(S,s) \simeq (P,p) \;\Leftrightarrow\; (S,s) \sqsubseteq (P,p) \;\&\; (P,p) \sqsubseteq (S,s)$$

REMARK. Let us denote by $\sim$ the (largest) bisimulation over process systems. Since a refinement over a process system is just a bisimulation, we have: if S is a process system then $p \sqsubseteq q \;\Leftrightarrow\; p \sim q$. Moreover if $S = (Q,\vartheta,!)$, and P is a process system (Q,κ) such that $! \subseteq \kappa \subseteq \vartheta$ then $(S,q) \sqsubseteq (P,q)$ for any $q \in Q$. Therefore any graphical specification is satisfied by some process.

Since we assumed image-finiteness, the refinement preorder $\sqsubseteq$ is, as usual, the limit of a sequence of approximations:

LEMMA 2.1. *Let S be a given specification. For each non-negative integer n, let $\sqsubseteq_n$ be the relation on states of S inductively given by:*

(i) $p \sqsubseteq_0 q$ *for all p and q*

(ii) $$p \sqsubseteq_{n+1} q \;\Leftrightarrow\; \begin{cases} p \xrightarrow[S]{a} p' \;\Rightarrow\; \exists q'.\, q \xrightarrow[S]{a} q' \;\&\; p' \sqsubseteq_n q' \\ q \xrightarrow[S]{a} q' \;\Rightarrow\; \exists p'.\, p \xrightarrow[S]{a} p' \;\&\; p' \sqsubseteq_n q' \end{cases}$$

Then $p \sqsubseteq q \;\Leftrightarrow\; \forall n.\; p \sqsubseteq_n q$.

The proof is standard (*cf.* [6,7]) □

For what regards our syntax, the refinement preorder is well-behaved:

LEMMA 2.2. *The refinement preorder over the system $\mathcal{G}$ is a precongruence, that is:*

$$t \sqsubseteq t' \;\Rightarrow\; \begin{cases} a.t \sqsubseteq a.t' \quad \text{and} \quad a!t \sqsubseteq a!t' \\ t+s \sqsubseteq t'+s \quad \text{and} \quad s+t \sqsubseteq s+t' \end{cases}$$

Moreover $\omega \sqsubseteq t$ and $t + \sum_{i\in I} a_i.t_i \sqsubseteq t \sqsubseteq t + 0$ for any term t.

The proof is straightforward. For instance $t + \sum_{i\in I} a_i.t_i \sqsubseteq t$ is true since $\sum_{i\in I} a_i.t_i$ does not require any transition. An obvious consequence of this property is $t + 0 \simeq t$. We could also prove that the sum is commutative and associative with respect to $\simeq$, a fact that justifies our notation for finite summations.

3. Logical Characterization.

In this section we introduce the logical specification framework, and we show the correspondence between refinement of graphical specifications and "refinement" – i.e. inclusion – of logical specifications. Moreover we shall show that the logical specification of any term of Θ may be expressed as a single characteristic formula.

The logic we use is Hennessy-Milner Logic, without negation (*cf.* [6,12,14]). The formulae are given by the grammar:

$$\phi \;::=\; \bot \mid \top \mid \langle a \rangle \phi \mid [\![a]\!]\phi \mid (\phi \wedge \phi) \mid (\phi \vee \phi)$$

We denote by Φ the set of formulae of this logic, and we shall use $\phi, \psi, \ldots$ to range over them.

We now define by induction on the formula ϕ the *satisfaction* relation $(S, s) \models \phi$, to be read as "the graphical specification (S, s) satisfies (or is a model of) the formula ϕ". This relation is defined as usual for the propositional connectives: $\bot$ and $\top$ stand respectively for false and true, $\phi \wedge \psi$ and $\phi \vee \psi$ are interpreted respectively as conjunction and disjunction. The interpretation of the modalities is somewhat unusual:

(i) an existential formula $\langle a \rangle \phi$ asserts the *existence* of an a-transition to a state satisfying ϕ. Therefore the interpretation of such a formula is given by means of the *required* transitions, namely:

$$(S, s) \models \langle a \rangle \phi \;\Leftrightarrow_{\text{def}}\; \exists s'.\; s \xrightarrow[S]{a} s' \;\&\; (S, s') \models \phi$$

(ii) a universal formula $[\![a]\!]\phi$ asserts that any *possible* a-transition reaches a state satisfying ϕ. Therefore its interpretation is given by means of the *allowed* transitions, thus:

$$(S, s) \models [\![a]\!]\phi \;\Leftrightarrow_{\text{def}}\; \forall s'.\; s \overset{a}{\underset{S}{\rightsquigarrow}} s' \;\Rightarrow\; (S, s') \models \phi$$

Note that if S is a process system, then this exactly the usual satisfaction relation. For what regards the graphical system over terms, we shall use the notation $t \models \phi$ for $(\mathcal{G}, t) \models \phi$. The notion of logical *consequence* (or implication) between formulae is defined by:

$$\phi \leq \psi \;\Leftrightarrow_{\text{def}}\; \forall (S, s).\; (S, s) \models \phi \;\Rightarrow\; (S, s) \models \psi$$

The logical equivalence associated with this preorder is

$$\phi \equiv \psi \;\Leftrightarrow\; \phi \leq \psi \;\&\; \psi \leq \phi$$

For instance $[\![a]\!]\top \equiv \top$ and $\langle a \rangle \bot \equiv \bot$. Note that the modalities $\langle a \rangle$ and $[\![a]\!]$ are not dual. For instance ω satisfies neither $[\![a]\!]\bot$ nor $\langle a \rangle \top$: the specification ω does not require anything, so it cannot satisfy a formula $\langle a \rangle \phi$. On the other hand $[\![a]\!]\bot$ expresses the fact that a system is not allowed to perform a, which is not the case for ω. Let us see some examples of formulae satisfied by the terms of Θ:

$$t \models \phi \;\Rightarrow\; \begin{cases} a.t \models [\![a]\!]\phi \quad \text{and} \quad \forall b \neq a \;\; a.t \models [\![b]\!]\bot \\ a!t \models \langle a \rangle \phi \quad \text{and} \quad a!t \models [\![a]\!]\phi \end{cases}$$

We now prove a generalized Hennessy-Milner Theorem, relating refinement of graphical specifications and logical implication. To this end, let us denote by $\mathcal{F}(S, s)$ the set of formulae satisfied by the graphical specification (S, s):

$$\mathcal{F}(S, s) =_{\text{def}} \{\phi \mid (S, s) \models \phi\}$$

This is the logical *theory* of the specification (S, s), or the set of formulae *expressed* by the specification (S, s).

Then our "Hennessy-Milner Theorem" is:

THEOREM 3.1. $(S,s) \sqsubseteq (P,p) \Leftrightarrow \mathcal{F}(S,s) \subseteq \mathcal{F}(P,p)$

PROOF: slight modification of the standard one, *cf.* [6,12,14]. We only sketch here the arguments. To prove that

$$(S,s) \sqsubseteq (P,p) \Rightarrow \mathcal{F}(S,s) \subseteq \mathcal{F}(P,p)$$

one uses lemma 2.1 and shows inductively on n that if $(S,s) \sqsubseteq_n (P,p)$ and (S,s) satisfies a formula ϕ whose modal depth is at most n, then (P,p) satisfies also this formula.

To prove the converse, assume $(S,s) \not\sqsubseteq_{n+1} (P,p)$. There are then two cases: either there exist $a \in A$ and s' such that $s \xrightarrow[S]{a} s'$ and for all p' if $p \xrightarrow[P]{a} p'$ then $(S,s') \not\sqsubseteq_n (P,p')$, or there exists $a \in A$ and p' such that $p \overset{a}{\underset{P}{\rightsquigarrow}} p'$ and for all s' if $s \overset{a}{\underset{S}{\rightsquigarrow}} s'$ then $(S,s') \not\sqsubseteq_n (P,p')$. In the first case one can find a formula of the form $\langle a \rangle(\phi_1 \wedge \cdots \wedge \phi_k)$ satisfied by (S,s) but not by (P,p), while in the second case a distinguishing formula exists whose shape is $[\![a]\!](\phi_1 \vee \cdots \vee \phi_k)$ ☐

REMARK. We could not maintain this result while including (classical) negation in the logic. Any theory $\mathcal{F}(S,s)$ of a specification is *complete* with respect to negation, that is for any formula ϕ either $\phi \in \mathcal{F}(S,s)$ or $\neg\phi \in \mathcal{F}(S,s)$. Therefore $\mathcal{F}(S,s) \subseteq \mathcal{F}(P,p) \Leftrightarrow \mathcal{F}(S,s) = \mathcal{F}(P,p)$.

This result allows us to explain why a system (P,p) is not a refinement of another one (S,s): in this case, there exists a formula satisfied by the specification (S,s), but not by the system (P,p). One may think of the theory $\mathcal{F}(S,s)$ as the *logical character* of the graphical system (S,s), and the previous result may be regarded as a representation theorem: a graphical specification may be *represented* by a logical specification (= set of formulae). More precisely, let us say that a set F of formulae represents a graphical specification (S,s) if and only if verifying that a system is a model (i.e. a refinement) of (S,s) amounts to check that this system is a model of F:

$$F \text{ represents } (S,s) \Leftrightarrow_{\text{def}} \forall (P,p).\ (S,s) \sqsubseteq (P,p) \Leftrightarrow \forall \phi \in F\ (P,p) \models \phi$$

Then the previous theorem asserts that $\mathcal{F}(S,s)$ represents (S,s). One may wonder whether this logical character may be condensed into a single *characteristic formula* $\chi(S,s)$. This problem has been addressed by Graf and Sifakis ([5], see also [13]). As a matter of fact, any term of Θ may be represented by a characteristic formula (*cf.* [10], where a more general result is obtained, by means of recursive formulae). More precisely, one may define a translation $\chi: \Theta \to \Phi$ yielding formulae $\chi(t)$ of the form

$$\bigwedge_{i \in I} \langle a_i \rangle \phi_i \wedge \bigwedge_{a \in A} [\![a]\!] \psi_a$$

The definition is as follows:

$$\chi(t) =_{\text{def}} \bigwedge_{\phi \in \delta(t)} \phi \wedge \bigwedge_{a \in A} [\![a]\!] \gamma_a(t)$$

(with the convention that $\top$ is the empty conjunction), where the set of formulae $\delta(t)$ – the existential part – and the formulae $\gamma_a(t)$ – the universal part w.r.t. a– are given inductively by:

(i) $\delta(0) = \emptyset$ & $\gamma_a(0) = \bot$ and $\delta(\omega) = \emptyset$ & $\gamma_a(\omega) = \top$

(ii) $\delta(a.t) = \emptyset$ and $\gamma_b(a.t) = \begin{cases} \chi(t) & \text{if } b = a \\ \bot & \text{otherwise} \end{cases}$

(iii) $\delta(a!t) = \{\langle a \rangle \chi(t)\}$ and $\gamma_b(a!t) = \gamma_b(a.t)$

(iv) $\delta(t_0 + t_1) = \delta(t_0) \cup \delta(t_1)$ and $\gamma_a(t_0 + t_1) = \gamma_a(t_0) \vee \gamma_a(t_1)$

Note that, due to the particular form of characteristic formulae we chose, the formula $\chi(\omega)$ cannot be simply $\top$ (but is logically equivalent to $\top$).

PROPOSITION 3.2. *For any term* $t \in \Theta$

$$(\mathcal{G}, t) \sqsubseteq (S, s) \;\Leftrightarrow\; (S, s) \models \chi(t)$$

The proof may be found in [10] ❑

REMARK. To deal with an infinite alphabet of actions, we should introduce more general modalities, namely $\bigwedge_{a \notin B} [\![a]\!]\bot$ where B is a cofinite set of actions.

As a consequence of this result, the refinement preorder on terms of Θ is represented by logical implication of their characteristic formulae:

COROLLARY 3.3. $t \sqsubseteq t' \;\Leftrightarrow\; \chi(t') \leq \chi(t)$ *and* $t \models \phi \;\Leftrightarrow\; \chi(t) \leq \phi$

4. Graphical Representation of Formulae.

In the previous section we have shown a correspondence from graphical specifications to the lattice of logical theories. In this section we will establish a converse correspondence. To this end, let us first state some definitions. A formula ϕ is *consistent* if it has a model, that is if there exists a graphical specification (S, s) such that $(S, s) \models \phi$. The formula ϕ is *prime* if, whenever it implies a disjunction, then it implies one of the disjuncts, that is:

$$\phi \text{ prime} \quad \Leftrightarrow_{\text{def}} \quad \phi \leq (\phi_0 \vee \phi_1) \;\Rightarrow\; \begin{cases} \phi \leq \phi_0 \\ \text{or} \\ \phi \leq \phi_1 \end{cases}$$

Usually a prime is required to be consistent, whereas $\bot$ is prime according to our definition. Let us see an example: for any $t \in \Theta$ the characteristic formula $\chi(t)$ is consistent since $t \models \chi(t)$; moreover if $\chi(t) \leq \phi \vee \psi$ then $t \models \phi \vee \psi$, that is, by definition of the satisfaction relation, $t \models \phi$ or $t \models \psi$, hence $\chi(t) \leq \phi$ or $\chi(t) \leq \psi$ (by the corollary 3.3). This shows that the characteristic formula of a term is a consistent prime formula. On the other hand, the (consistent) formula $\phi = \langle a \rangle \top \vee \langle b \rangle \top$ is an example of non-prime formula: neither $\phi_0 = \langle a \rangle \top$ nor $\phi_1 = \langle b \rangle \top$ is a consequence of ϕ, since $a!\omega$ (resp. $b!\omega$) is a model of ϕ but not of ϕ_1 (resp. ϕ_0).

We shall say that a formula ϕ is *graphically representable* if and only if there exists a graphical specification (S, s) such that checking that a system is a model of ϕ amounts to verifying that this system is a refinement of (S, s), thus:

$$(S, s) \text{ represents } \phi \;\Leftrightarrow_{\text{def}}\; \forall (P, p).\ (P, p) \models \phi \;\Leftrightarrow\; (S, s) \sqsubseteq (P, p)$$

Obviously a system represents a formula if and only if the formula represents this system (in the sense of the previous section), and if (S, s) represents ϕ then $(S, s) \models \phi$ (hence ϕ is consistent). Then our main result is the following:

THEOREM. *A formula is graphically representable if and only if it is consistent and prime.*

More precisely we shall show that any formula is representable by a finite set of terms of Θ, and this set can be reduced to a singleton when the given formula is consistent and prime. This establishes a Galois connection between the preordered set of terms $(\Theta, \sqsubseteq)$ and the preordered set $(\Pi, \leq)$ of consistent prime formulae.

The proof of our result relies on a specific notion of "normal form" for formulae. Let us first introduce the usual notion of normal form – using the convention that $\bot$ is the empty disjunction and $\top$ the empty conjunction. A formula is said to be in *normal form* if it has the form

$$\bigvee_{i \in I} \Big(\bigwedge_{j \in J_i} \langle a^i_j \rangle \phi^i_j \wedge \bigwedge_{k \in K_i} [\![b^i_k]\!] \psi^i_k \Big)$$

where ϕ^i_j, ψ^i_k are themselves in normal form. For instance the characteristic formulae $\chi(t)$ of terms are in normal form, without head disjunction. Using standard distributivity properties (and tautologies like $\phi \wedge \top \equiv \phi$), it is easy to see that:

FACT. *For every formula of Φ there exists a logically equivalent formula in normal form, with the same modal depth.*

We will now prove a stronger result, introducing our more specific notion of normal form – which is not purely syntactic. Note that in a formula in normal form the universal modalities may be grouped (within a disjunct), so that for any action a there is exactly one subformula $[\![a]\!]\psi$ in each disjunct (recall that $[\![a]\!]\top \equiv \top$, hence these formulae may be freely added, when the set of actions is finite). Also one may ensure that if a disjunct contains subformulae $[\![a]\!]\psi$ and $\langle a\rangle\phi$, then the latter "includes" the universal information about a-transitions, that is $\phi \leq \psi$. Therefore we shall say that a formula is in *strong normal form* if it is $\top$ or has the form

$$\bigvee_{i\in I}\Big(\bigwedge_{j\in J_i}\langle a^i_j\rangle\phi^i_j \wedge \bigwedge_{a\in A}[\![a]\!]\psi^i_a\Big)$$

where ϕ^i_j, ψ^i_a are in strong normal form, and

$$a^i_j = a \;\Rightarrow\; \phi^i_j \leq \psi^i_a$$

Note that due to our conventions, $\bot$ is in strong normal form. Moreover, one may check that the characteristic formulae of terms are in strong normal form.

LEMMA 4.1. *For every formula ϕ of Φ there exists a logically equivalent formula ϕ^* in strong normal form, with the same modal depth.*

PROOF: by induction on the modal depth of ϕ. By the previous fact, we may assume that ϕ is in normal form, that is

$$\phi = \bigvee_{i\in I}\Big(\bigwedge_{j\in J_i}\langle a^i_j\rangle\phi^i_j \wedge \bigwedge_{k\in K_i}[\![b^i_k]\!]\psi^i_k\Big)$$

Let

$$\psi^i_a = \bigwedge_{\{k\,|\,b^i_k = a\}}\psi^i_k$$

so that $\psi^i_a = \top$ if $\{k \mid b^i_k = a\} = \emptyset$, and

$$\varphi^i_j = (\phi^i_j \wedge \psi^i_a) \quad \text{where } a = a^i_j$$

By induction hypothesis, both the formulae ψ^i_a and φ^i_j possess equivalent strong normal forms, respectively $(\psi^i_a)^*$ and $(\varphi^i_j)^*$. Now if we let

$$\phi^* = \bigvee_{i\in I}\Big(\bigwedge_{j\in J_i}\langle a^i_j\rangle(\varphi^i_j)^* \wedge \bigwedge_{a\in A}[\![a]\!](\psi^i_a)^*\Big)$$

it is easy to see that $\phi \equiv \phi^*$, due to the basic laws

$$[\![a]\!]\psi \wedge [\![a]\!]\gamma \equiv [\![a]\!](\psi\wedge\gamma) \qquad \text{and} \qquad [\![a]\!]\top \equiv \top$$

$$\langle a\rangle\psi \wedge [\![a]\!]\gamma \equiv \langle a\rangle(\psi\wedge\gamma) \wedge [\![a]\!]\gamma$$

To prove this last equation, one uses $\xrightarrow{a} \subseteq \overset{a}{\rightsquigarrow}$. Moreover it is easy to see that ϕ^* is in strong normal form, since $(\psi\wedge\gamma) \leq \gamma$ □

Let us say that a formula ϕ is represented by a set T of terms if the following holds:

$$\forall (S,s).\quad (S,s) \models \phi \;\Leftrightarrow\; \exists t \in T.\quad (\mathcal{G},t) \sqsubseteq (S,s)$$

We then define, for a formula ϕ in strong normal form, a finite set $\theta(\phi)$ of terms which is intended to represent ϕ. The definition is by induction on the structure of ϕ:

(i) $\theta(\top) = \{\omega\}$

(ii) let $\phi = \bigwedge_{j\in J} \langle a_j\rangle \phi_j \wedge \bigwedge_{a\in A} [\![a]\!]\psi_a$ then

$$t \in \theta(\phi) \;\Leftrightarrow_{\text{def}}\; \forall j \in J\ \exists u_j \in \theta(\phi_j)\quad t = \sum_{j\in J} a_j ! u_j + \sum_{a\in A} \Big(\sum_{v\in\theta(\psi_a)} a.v \Big)$$

(iii) $\theta(\bigvee_{i\in I} \phi_i) = \bigcup_{i\in I} \theta(\phi_i)$

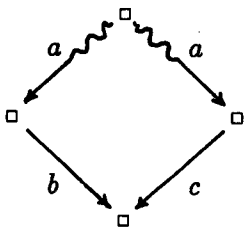
Note that in the second case, if $\theta(\phi_j) = \emptyset$ for some $j \in J$ then $\theta(\phi) = \emptyset$. As we shall see, this means $\phi_j \equiv \bot$, hence $\phi \equiv \bot$.

EXAMPLES. This definition suggests, to cope with the subterms $\sigma_B = \sum_{a\in B} a.\omega$ arising from the formulae $[\![a]\!]\top$, the following *graphical convention*: in drawings, a square $\square$ represents a state s such that $s \overset{a}{\rightsquigarrow} \omega$ for any action a that is not explicitly drawn as outgoing from this state. Then for instance ω is simply represented by $\square$ (without any exiting edge). The set of terms representing $\phi = \langle a\rangle\top$ consists of the single term $a!\omega + \sigma_A$. Using our graphical convention, the specification determined by this term may be drawn:

$$\square \xrightarrow{a} \square$$

Therefore the formula $\langle b\rangle\top \vee \langle c\rangle\top$ is represented by a set of two such graphical specifications (as we have seen, this formula is not prime).

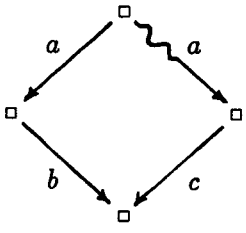
Similarly, the graphical interpretation of the formula $\psi = [\![a]\!](\langle b\rangle\top \vee \langle c\rangle\top)$ consists of the single term $\sigma_{A-\{a\}} + a.(b!\omega + \sigma_A) + a.(c!\omega + \sigma_A)$. The graphical specification determined by this term may be drawn as:



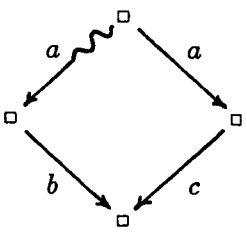
Note that the formula $\phi \wedge \psi$ is not in strong normal form, but is equivalent to

$$(\langle a\rangle\langle b\rangle\top \wedge \psi) \;\vee\; (\langle a\rangle\langle c\rangle\top \wedge \psi)$$

(since $\langle a\rangle(\phi_0 \vee \phi_1) \equiv \langle a\rangle\phi_0 \vee \langle a\rangle\phi_1$). This formula, which is easily converted into a strong normal form by adding $[\![b]\!]\top$ and $[\![c]\!]\top$ (assuming $A = \{a,b,c\}$), may be represented by a set of two graphical specifications:



and



We can now prove our claim that for any formula ϕ in strong normal form, the set $\theta(\phi)$ represents this formula:

PROPOSITION 4.2. *For any formula ϕ in strong normal form*

$$\forall (S,s).\quad (S,s) \models \phi \;\Leftrightarrow\; \exists t \in \theta(\phi).\quad (\mathcal{G},t) \sqsubseteq (S,s)$$

PROOF: first we show that $t \in \theta(\phi) \;\Rightarrow\; t \models \phi$. This will show the "$\Leftarrow$" part of the proposition (using the Hennessy-Milner Theorem). We proceed by induction on the structure of ϕ. The case $\phi = \top$ is trivial, so let us assume that

$$\phi = \bigwedge_{j\in J} \langle a_j\rangle \phi_j \wedge \bigwedge_{a\in A} [\![a]\!]\psi_a$$

Then

$$t = \sum_{j\in J} a_j!u_j + \sum_{a\in A}\Big(\sum_{v\in\theta(\psi_a)} a.v\Big)$$

where $u_j \in \theta(\phi_j)$ for any $j \in J$. If $t \overset{a}{\rightsquigarrow} t'$ then either $t' \in \theta(\psi_a)$ or $t \in \theta(\phi_j)$ for some $j \in J$ such that $a_j = a$. In the first case we have by induction hypothesis $t' \models \psi_a$, and similarly in the second case $t' \models \phi_j$. Since ϕ is in strong normal form we have $\phi_j \leq \psi_a$, hence in any case $t' \models \psi_a$. This shows:

$$\forall a \in A \quad t \models [\![a]\!]\psi_a$$

On the other hand we have

$$\forall j \in J \quad t \models \langle a_j\rangle \phi_j$$

since by definition $t \xrightarrow{a_j} u_j$ with $u_j \in \theta(\phi_j)$, and $u_j \models \phi_j$ by induction hypothesis. By definition of the satisfaction relation as regards conjunction, we then have $t \models \phi$. To conclude this first part of the proof, we note that the case $\phi = \bigvee_{i\in I} \phi_i$ is trivial since $\phi_i \leq \bigvee_{i\in I} \phi_i$.

Conversely assume that $(S,s) \models \phi$ (ϕ in strong normal form). We show by induction on the structure of the formula ϕ that there exists $t \in \theta(\phi)$ such that $(\mathcal{G},t) \sqsubseteq (S,s)$. The disjunctive case, i.e. $\phi = \bigvee_{i\in I} \phi_i$, is trivial, as well as the case $\phi = \top$, so let us assume that

$$\phi = \bigwedge_{j\in J} \langle a_j\rangle \phi_j \wedge \bigwedge_{a\in A} [\![a]\!]\psi_a$$

Since $(S,s) \models \langle a_j\rangle \phi_j$, by induction hypothesis for every $j \in J$ there exists q_j such that $s \xrightarrow[S]{a_j} q_j$ and $u_j \in \theta(\phi_j)$ such that $(\mathcal{G},u_j) \sqsubseteq (S,q_j)$. Therefore we may define a "ϕ-approximation" of (S,s):

$$t = \sum_{j\in J} a_j!u_j + \sum_{a\in A}\Big(\sum_{v\in\theta(\psi_a)} a.v\Big)$$

Clearly $t \in \theta(\phi)$. Let us show that $(\mathcal{G},t) \sqsubseteq (S,s)$: if $t \xrightarrow{a} t'$ then there exists $j \in J$ such that $a = a_j$ and $t' = u_j$, and $(\mathcal{G},u_j) \sqsubseteq (S,q_j)$ by construction. Therefore a transition required by t is matched by a definite transition of (S,s). Now if $s \overset{a}{\underset{S}{\rightsquigarrow}} s'$ then $(S,s') \models \psi_a$ (for $(S,s) \models [\![a]\!]\psi_a$), hence by induction hypothesis there exists $v \in \theta(\psi_a)$ such that $(\mathcal{G},v) \sqsubseteq (S,s')$, and $t \overset{a}{\rightsquigarrow} v$ by definition of the term t □

REMARK. To deal with a possibly infinite set of actions, we should restrict the notion of strong normal form, allowing only finite universal parts $\bigwedge_{b\in B} [\![b]\!]\psi_b$ in each disjunct, and use terms of the form $t + \sigma_{(A-B)}$ to represent these formulae.

One may note that this result shows that any formula is logically equivalent to a (possibly empty) disjunction of consistent prime formulae, namely:

$$\phi \equiv \bigvee_{t \in \theta(\phi^*)} \chi(t)$$

where ϕ^* is a formula in strong normal form which is logically equivalent to ϕ (lemma 4.1). Remark also that our proofs entail the decidability of the satisfiability problem in our logic: ϕ is satisfiable if and only if $\theta(\phi^*)$ is non-empty, where ϕ^* is built as in the proof of lemma 4.1. Moreover the consequence relation $\phi \leq \psi$ is also decidable since

$$\phi \leq \psi \;\Leftrightarrow\; \forall t \in \theta(\phi^*)\; \exists t' \in \theta(\psi^*).\; t' \sqsubseteq t$$

We may now prove our main result:

PROOF of the THEOREM: first we show that if ϕ is graphically represented by (S, s), then ϕ is consistent (this is obvious since $(S, s) \models \phi$) and prime. Assume that $\phi \leq (\phi_0 \vee \phi_1)$; then $(S, s) \models \phi_0 \vee \phi_1$ that is, by definition of the satisfaction relation, $(S, s) \models \phi_i$ for $i = 0$ or $i = 1$. Since (S, s) represents ϕ, for any model (P, p) of ϕ we then have, using the Hennessy-Milner Theorem, $(P, p) \models \phi_i$. This shows $\phi \leq \phi_i$.

Now assume that ϕ is consistent and prime, and let $\theta(\phi^*) = \{t_1, \ldots, t_n\}$ where ϕ^* is a formula in strong normal form which is logically equivalent to ϕ (lemma 4.1). Since ϕ is consistent (hence also ϕ^*), this set is non-empty. If $n = 1$ then t_1 represents ϕ^*, by the previous proposition, and also ϕ since $\phi \equiv \phi^*$. If $n > 1$, then $\phi \leq \chi(t_i)$ for some i since $\phi \equiv \bigvee_{1 \leq i \leq n} \chi(t_i)$ and ϕ is prime. But then $\phi \equiv \chi(t_i)$ since $\chi(t_i) \leq \phi$ (due to the general law $\psi \leq \psi \vee \gamma$), therefore:

$$\begin{aligned} (S, s) \models \phi &\;\Leftrightarrow\; (S, s) \models \chi(t_i) \quad \text{for } \chi(t_i) \equiv \phi \\ &\;\Leftrightarrow\; (\mathcal{G}, t_i) \sqsubseteq (S, s) \quad \text{by the proposition 3.2} \end{aligned}$$

that is: t_i represents ϕ □

REFERENCES

[1] A. ARNOLD, *MEC: a System for Constructing and Analysing Transition Systems*, to appear in Proceedings of the 1st Workshop on Automatic Verification Methods for Finite State Systems, Grenoble (1989).

[2] G. BOUDOL, R. DE SIMONE, D. VERGAMINI, *Experiment with* AUTO *and* AUTOGRAPH *on a Simple Case of Sliding Window Protocol*, INRIA Res. Rep. 870 (1988).

[3] R. CLEAVELAND, J. PARROW, B. STEFFEN, *The Concurrency Worbench*, Rep. ECS-LFCS-89-83, Edinburgh (1989).

[4] J. GODSKESEN, K.G. LARSEN, M. ZEEBERG, TAV *Users Manual*, Ålborg University (1989).

[5] S. GRAF, J. SIFAKIS, *A Logic for the Description of Non-deterministic Programs and their Properties*, Information and Control 68 (1986) 254-270.

[6] M. HENNESSY, R. MILNER, *Algebraic Laws for Nondeterminism and Concurrency*, JACM 32 (1985) 137-161.

[7] H. HÜTTEL, *Operational and Denotational Properties of a Modal Process Logic*, Report 88-27, Dept. of Mathematics and Comput. Sci., Ålborg University (1988).

[8] H. HÜTTEL, K. G. LARSEN, *The Use of Static Constructs in a Modal Process Logic*, Logic at Botik 89, Lecture Notes in Comput. Sci. 363 (1989) 163-180.

[9] K. G. LARSEN, B. THOMSEN, *A Modal Process Logic*, Proc. LICS 88 (1988) 203-210.

[10] K. G. LARSEN, *Modal Specifications*, Report 89-9, Dept. of Mathematics and Comput. Sci., Ålborg University (1988) to appear in Proceedings of the 1st Workshop on Automatic Verification Methods for Finite State Systems, Grenoble.

[11] V. LECOMPTE, E. MADELAINE, D. VERGAMINI, AUTO, *un système de vérification de processus parallèles et communicants*, Rapport Technique INRIA 83 (1987).

[12] R. MILNER, *A Modal Characterisation of Observable Machine-Behaviour*, CAAP 81, Lecture Notes in Comput. Sci. 112 (1981) 25-34.

[13] A. PNUELI, *Linear and Branching Structures in the Semantics and Logics of Reactive Systems*, ICALP 85, Lecture Notes in Comput. Sci. 194 (1985) 15-32.

[14] C. STIRLING, *Modal Logics for Communicating Systems*, Theoretical Comput. Sci. 49 (1987) 311-347.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique